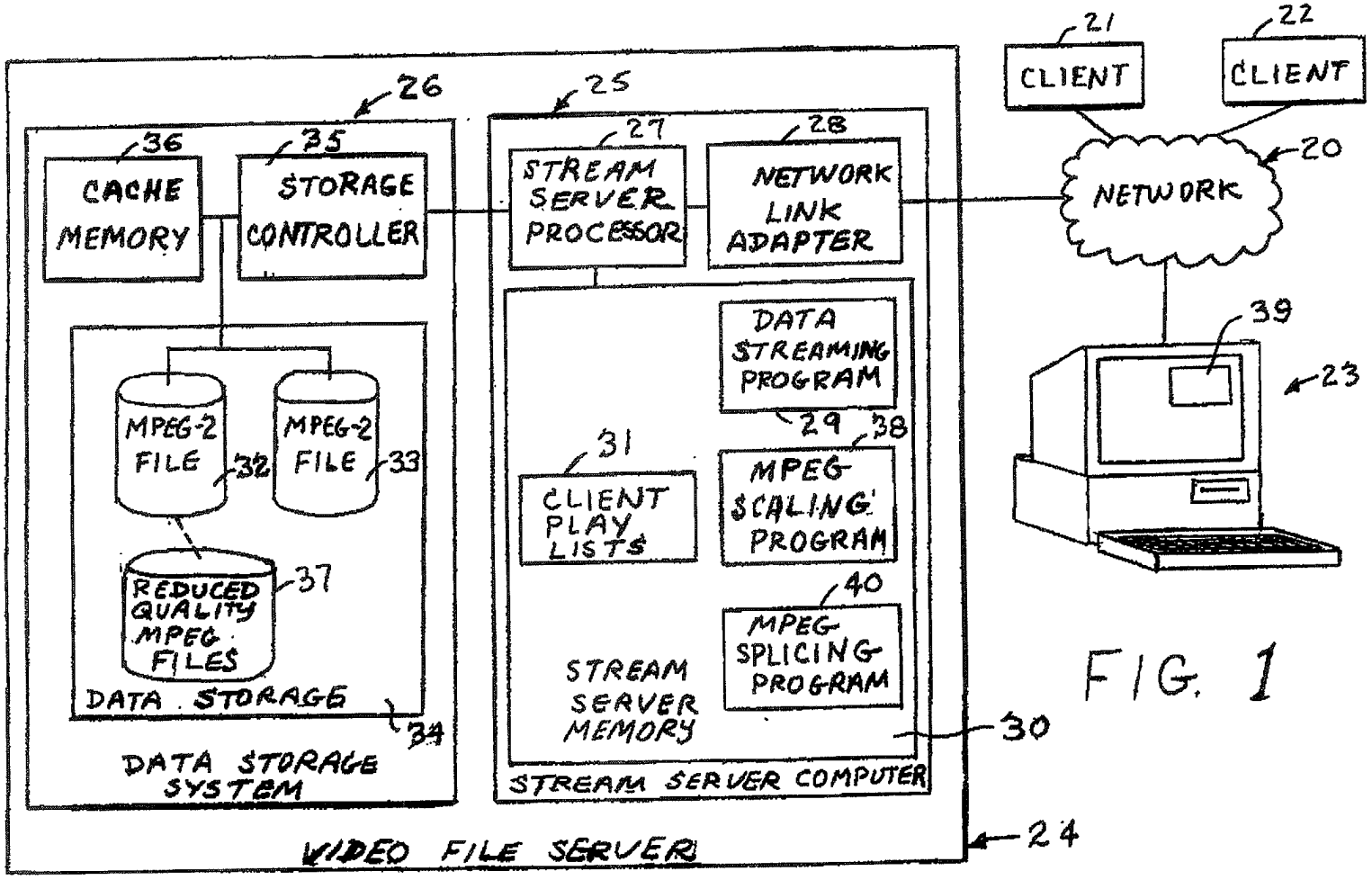
21
CLIENT
22
CLIENT
20
NETWORK
39
23
26
25
36
CACHE MEMORY
35
STORAGE CONTROLLER
27
STREAM SERVER PROCESSOR
28
NETWORK LINK ADAPTER
DATA STREAMING PROGRAM
29
38
MPEG SCALING PROGRAM
31
CLIENT PLAY LISTS
MPEG-2 FILE
32
MPEG-2 FILE
33
REDUCED QUALITY MPEG FILES
37
40
MPEG SPLICING PROGRAM
STREAM SERVER MEMORY
30
DATA STORAGE
34
DATA STORAGE SYSTEM
STREAM SERVER COMPUTER
VIDEO FILE SERVER
24

FIG. 1

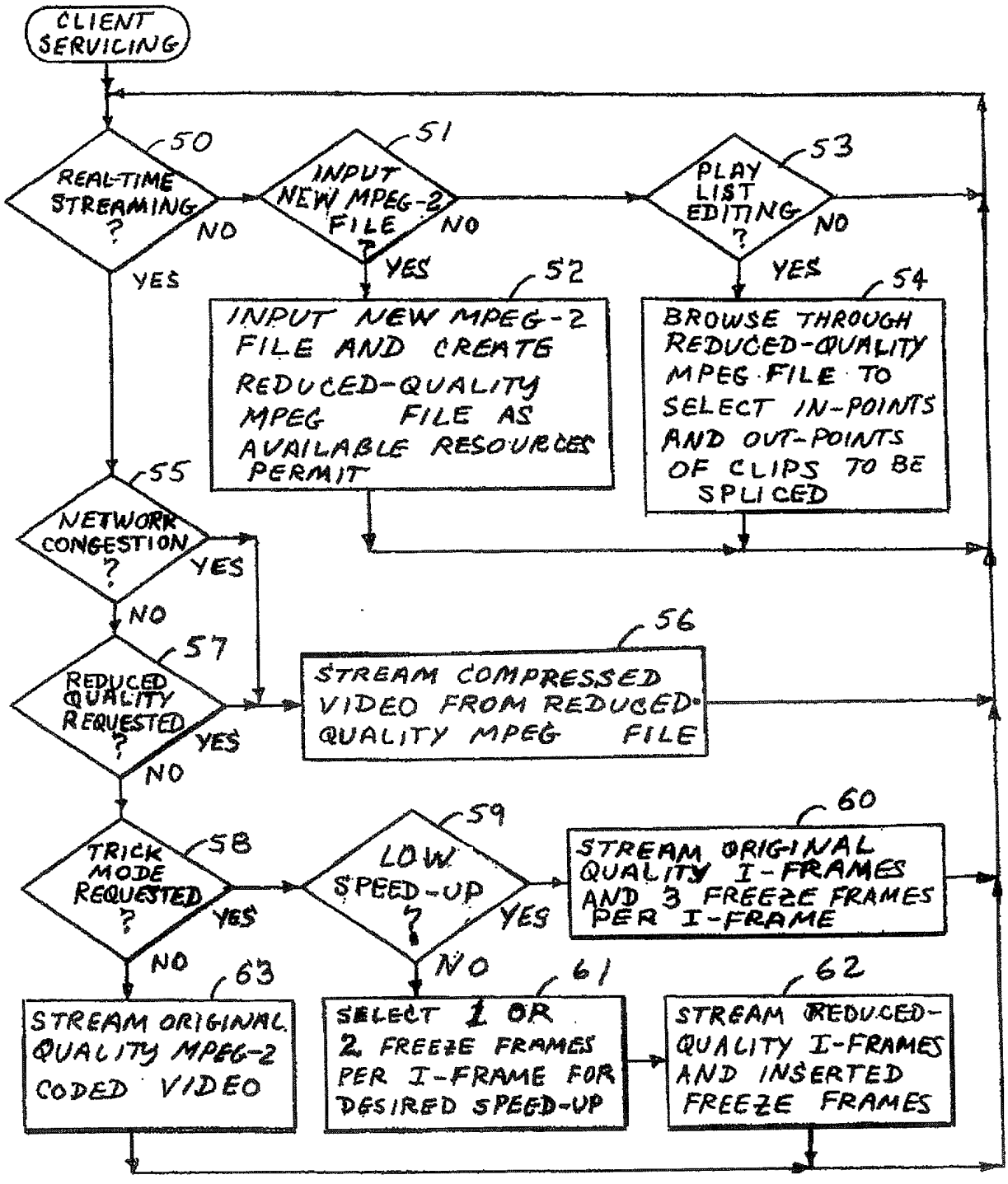
CLIENT SERVICING
50
REAL-TIME STREAMING ?
NO
YES
51
INPUT NEW MPEG-2 FILE ?
NO
YES
52
INPUT NEW MPEG-2 FILE AND CREATE REDUCED-QUALITY MPEG FILE AS AVAILABLE RESOURCES PERMIT
53
PLAY LIST EDITING ?
NO
YES
54
BROWSE THROUGH REDUCED-QUALITY MPEG FILE TO SELECT IN-POINTS AND OUT-POINTS OF CLIPS TO BE SPLICED
55
NETWORK CONGESTION ?
YES
NO
56
STREAM COMPRESSED VIDEO FROM REDUCED-QUALITY MPEG FILE
57
REDUCED QUALITY REQUESTED ?
YES
NO
58
TRICK MODE REQUESTED ?
YES
NO
59
LOW SPEED-UP ?
YES
NO
60
STREAM ORIGINAL QUALITY I-FRAMES AND 3 FREEZE FRAMES PER I-FRAME
61
SELECT 1 OR 2 FREEZE FRAMES PER I-FRAME FOR DESIRED SPEED-UP
62
STREAM REDUCED-QUALITY I-FRAMES AND INSERTED FREEZE FRAMES
63
STREAM ORIGINAL QUALITY MPEG-2 CODED VIDEO

FIG. 2

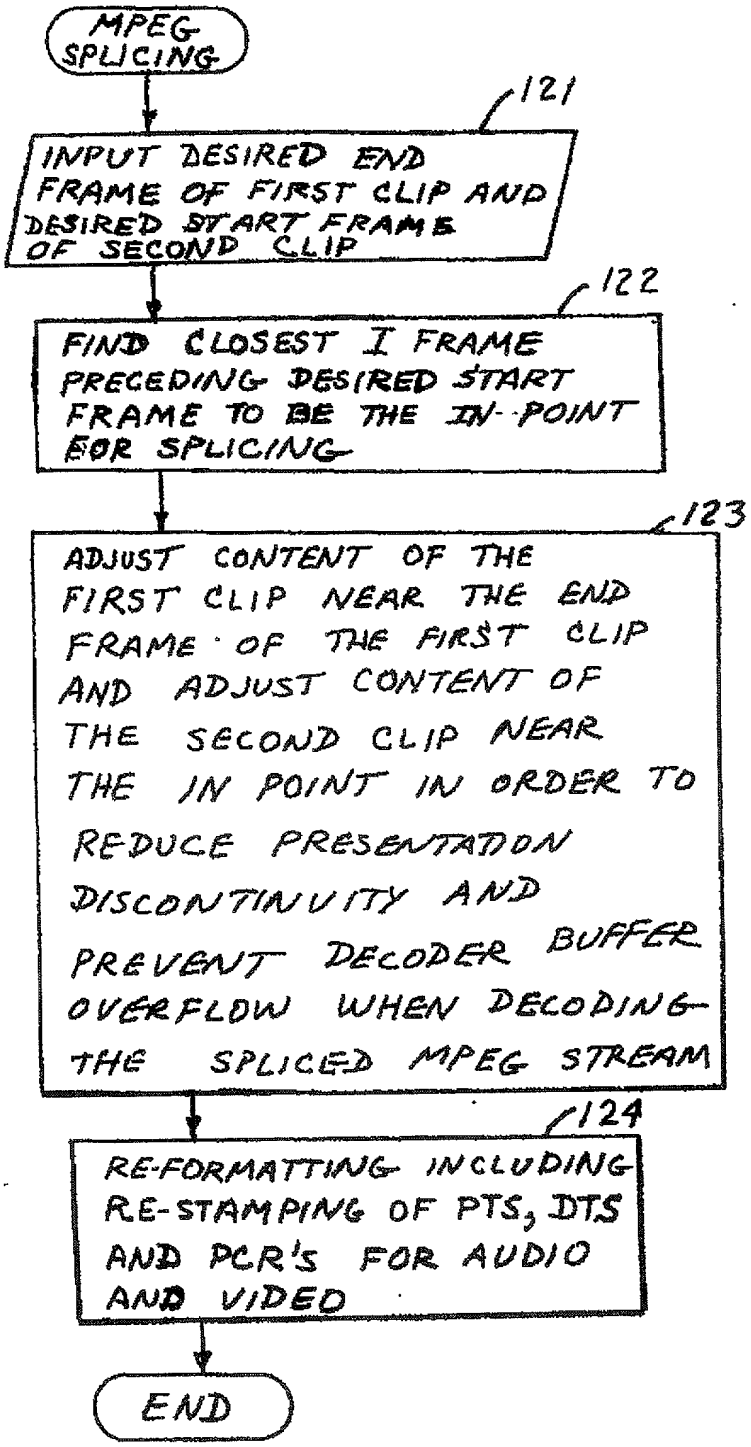
MPEG SPLICING
121
INPUT DESIRED END FRAME OF FIRST CLIP AND DESIRED START FRAME OF SECOND CLIP
122
FIND CLOSEST I FRAME PRECEDING DESIRED START FRAME TO BE THE IN-POINT FOR SPLICING
123
ADJUST CONTENT OF THE FIRST CLIP NEAR THE END FRAME OF THE FIRST CLIP AND ADJUST CONTENT OF THE SECOND CLIP NEAR THE IN POINT IN ORDER TO REDUCE PRESENTATION DISCONTINUITY AND PREVENT DECODER BUFFER OVERFLOW WHEN DECODING THE SPLICED MPEG STREAM
124
RE-FORMATTING INCLUDING RE-STAMPING OF PTS, DTS AND PCR's FOR AUDIO AND VIDEO
END

FIG. 3

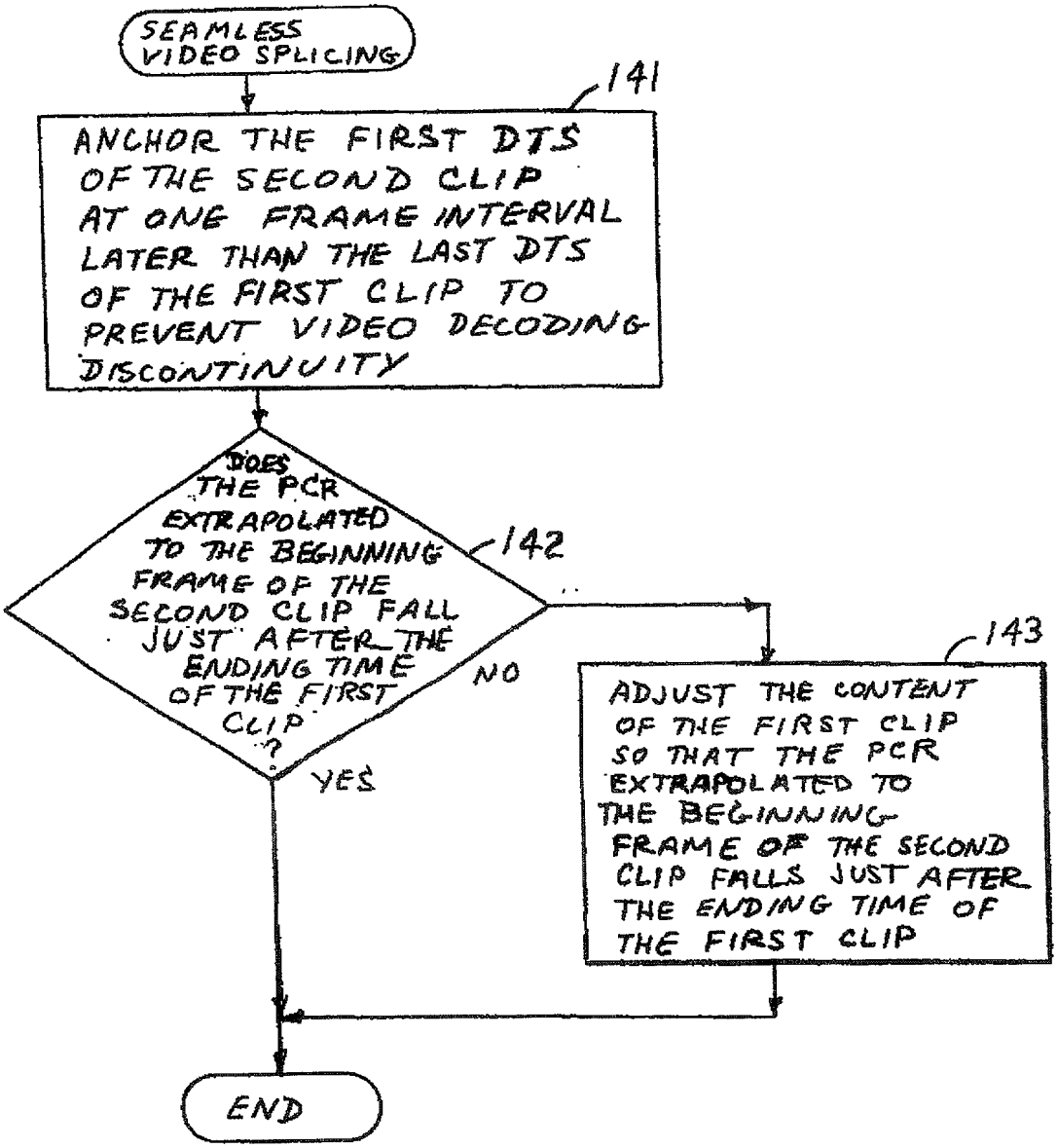
SEAMLESS VIDEO SPLICING
141
ANCHOR THE FIRST DTS OF THE SECOND CLIP AT ONE FRAME INTERVAL LATER THAN THE LAST DTS OF THE FIRST CLIP TO PREVENT VIDEO DECODING DISCONTINUITY
DOES THE PCR EXTRAPOLATED TO THE BEGINNING FRAME OF THE SECOND CLIP FALL JUST AFTER THE ENDING TIME OF THE FIRST CLIP ?
142
NO
YES
143
ADJUST THE CONTENT OF THE FIRST CLIP SO THAT THE PCR EXTRAPOLATED TO THE BEGINNING FRAME OF THE SECOND CLIP FALLS JUST AFTER THE ENDING TIME OF THE FIRST CLIP
END

FIG. 4

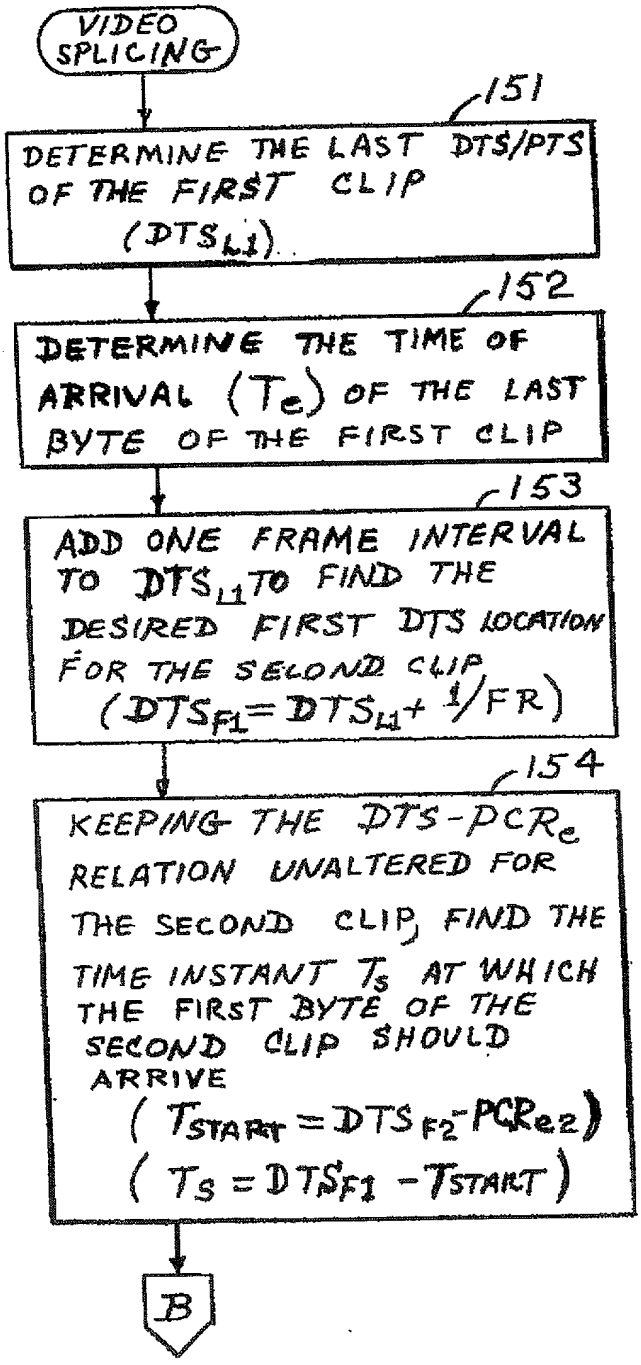
VIDEO SPLICING
151
DETERMINE THE LAST DTS/PTS OF THE FIRST CLIP (DTS_{L1})
152
DETERMINE THE TIME OF ARRIVAL (T_e) OF THE LAST BYTE OF THE FIRST CLIP
153
ADD ONE FRAME INTERVAL TO DTS_{L1} TO FIND THE DESIRED FIRST DTS LOCATION FOR THE SECOND CLIP ($DTS_{F1} = DTS_{L1} + 1/FR$)
154
KEEPING THE $DTS-PCR_e$ RELATION UNALTERED FOR THE SECOND CLIP, FIND THE TIME INSTANT T_s AT WHICH THE FIRST BYTE OF THE SECOND CLIP SHOULD ARRIVE
($T_{START} = DTS_{F2} - PCR_{e2}$)
($T_s = DTS_{F1} - T_{START}$)
B

FIG. 5

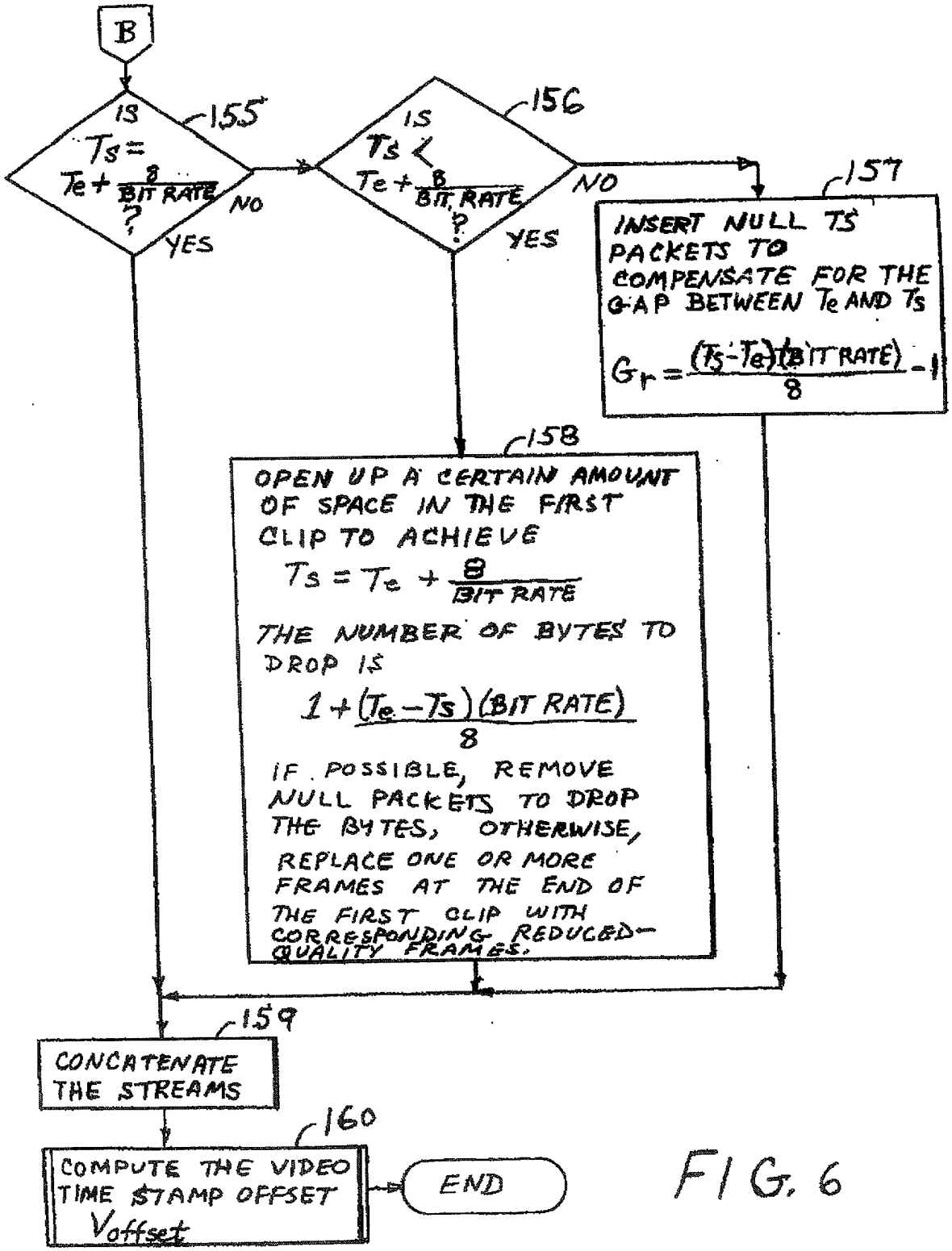
B
IS $Ts = Te + \frac{8}{BIT\ RATE}$?
155
NO
YES
IS $Ts < Te + \frac{8}{BIT\ RATE}$?
156
NO
YES
157
INSERT NULL TS PACKETS TO COMPENSATE FOR THE GAP BETWEEN Te AND Ts
$G_r = \frac{(Ts - Te)(BIT\ RATE)}{8} - 1$
158
OPEN UP A CERTAIN AMOUNT OF SPACE IN THE FIRST CLIP TO ACHIEVE
$Ts = Te + \frac{8}{BIT\ RATE}$
THE NUMBER OF BYTES TO DROP IS
$1 + \frac{(Te - Ts)(BIT\ RATE)}{8}$
IF POSSIBLE, REMOVE NULL PACKETS TO DROP THE BYTES, OTHERWISE, REPLACE ONE OR MORE FRAMES AT THE END OF THE FIRST CLIP WITH CORRESPONDING REDUCED-QUALITY FRAMES.
159
CONCATENATE THE STREAMS
160
COMPUTE THE VIDEO TIME STAMP OFFSET Voffset
END

FIG. 6

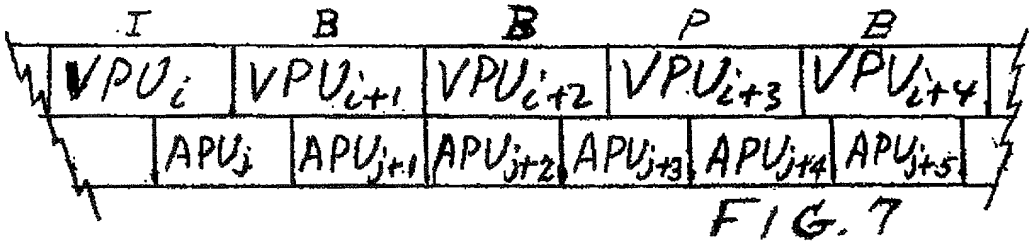

FIG. 7

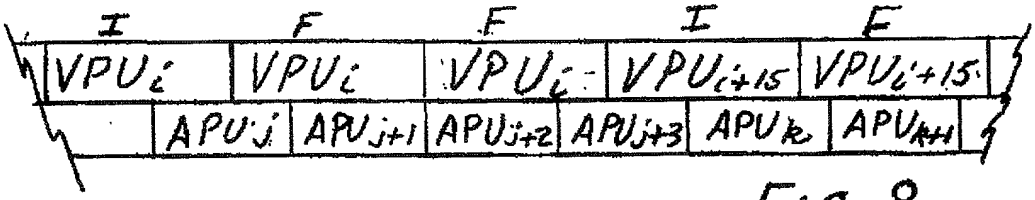

FIG. 8

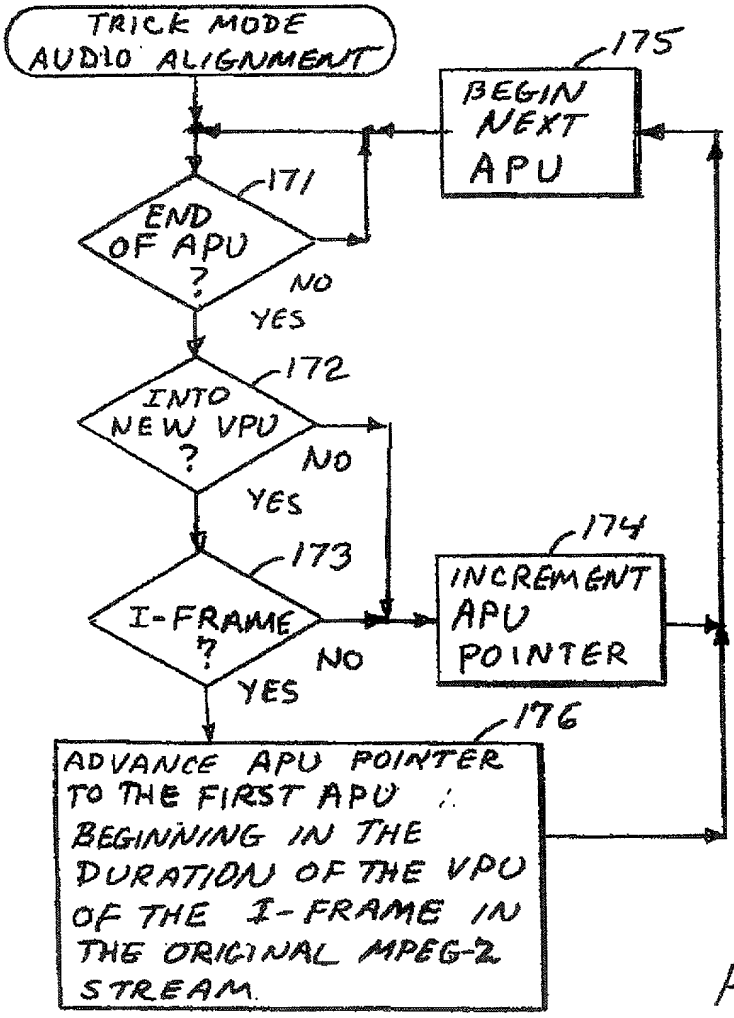

FIG. 9

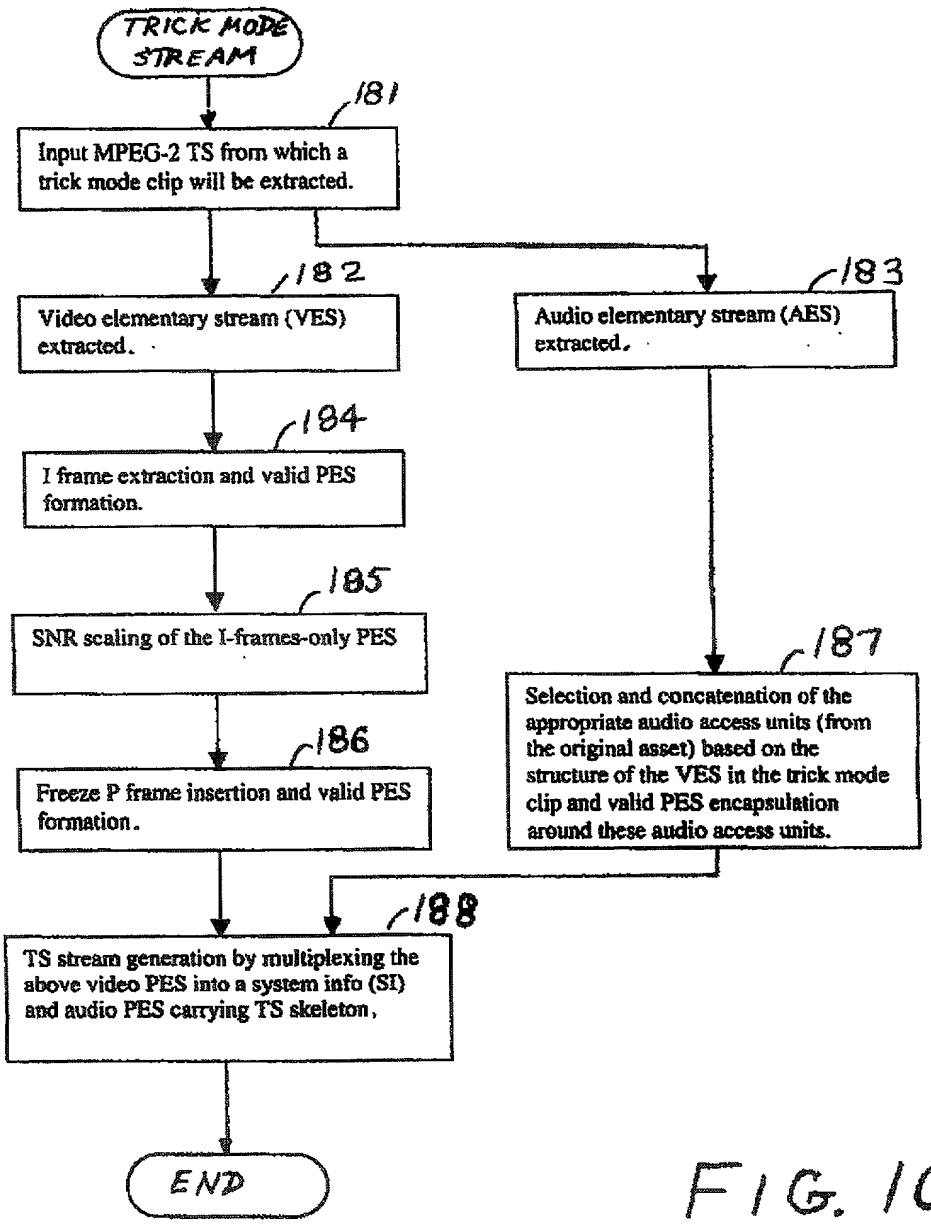
TRICK MODE STREAM
181
Input MPEG-2 TS from which a trick mode clip will be extracted.
182
Video elementary stream (VES) extracted.
183
Audio elementary stream (AES) extracted.
184
I frame extraction and valid PES formation.
185
SNR scaling of the I-frames-only PES
186
Freeze P frame insertion and valid PES formation.
187
Selection and concatenation of the appropriate audio access units (from the original asset) based on the structure of the VES in the trick mode clip and valid PES encapsulation around these audio access units.
188
TS stream generation by multiplexing the above video PES into a system info (SI) and audio PES carrying TS skeleton.
END

FIG. 10

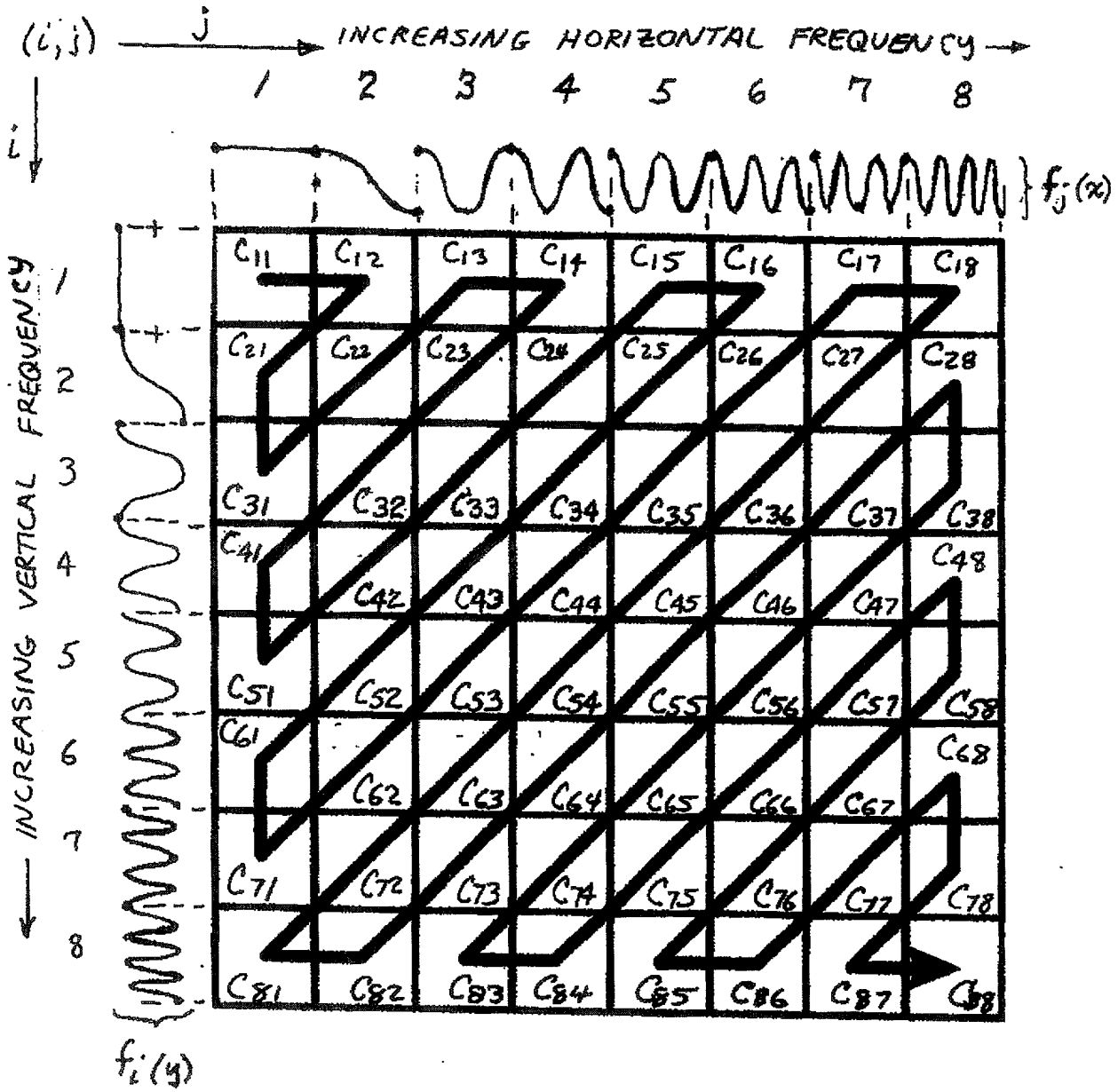

FIG. 11
(PRIOR ART)

0082271 "1550260

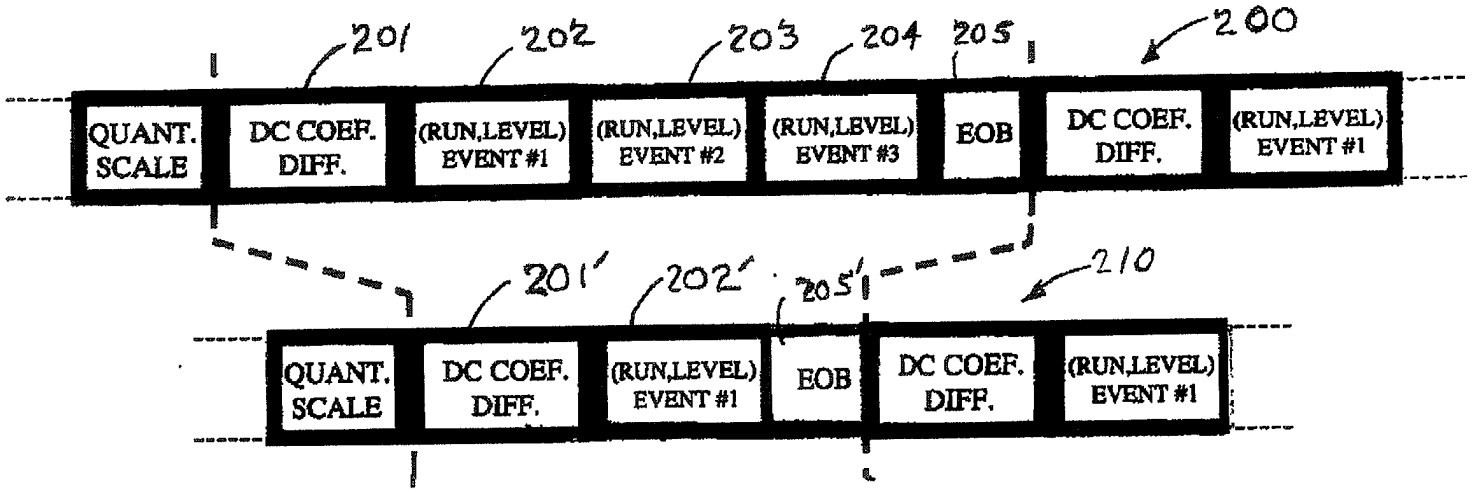

FIG. 12

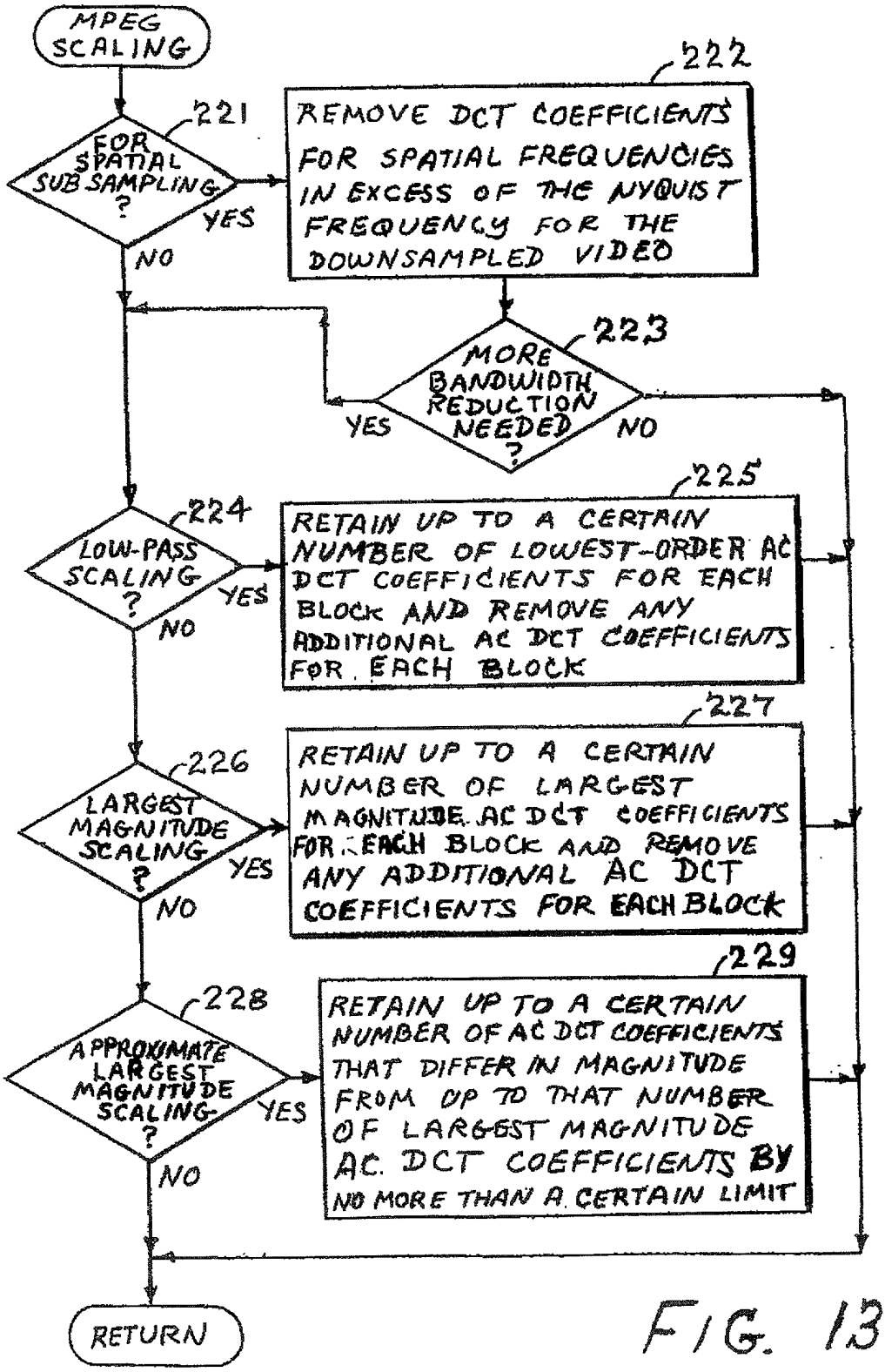

FIG. 13

Document: US 5,956,674

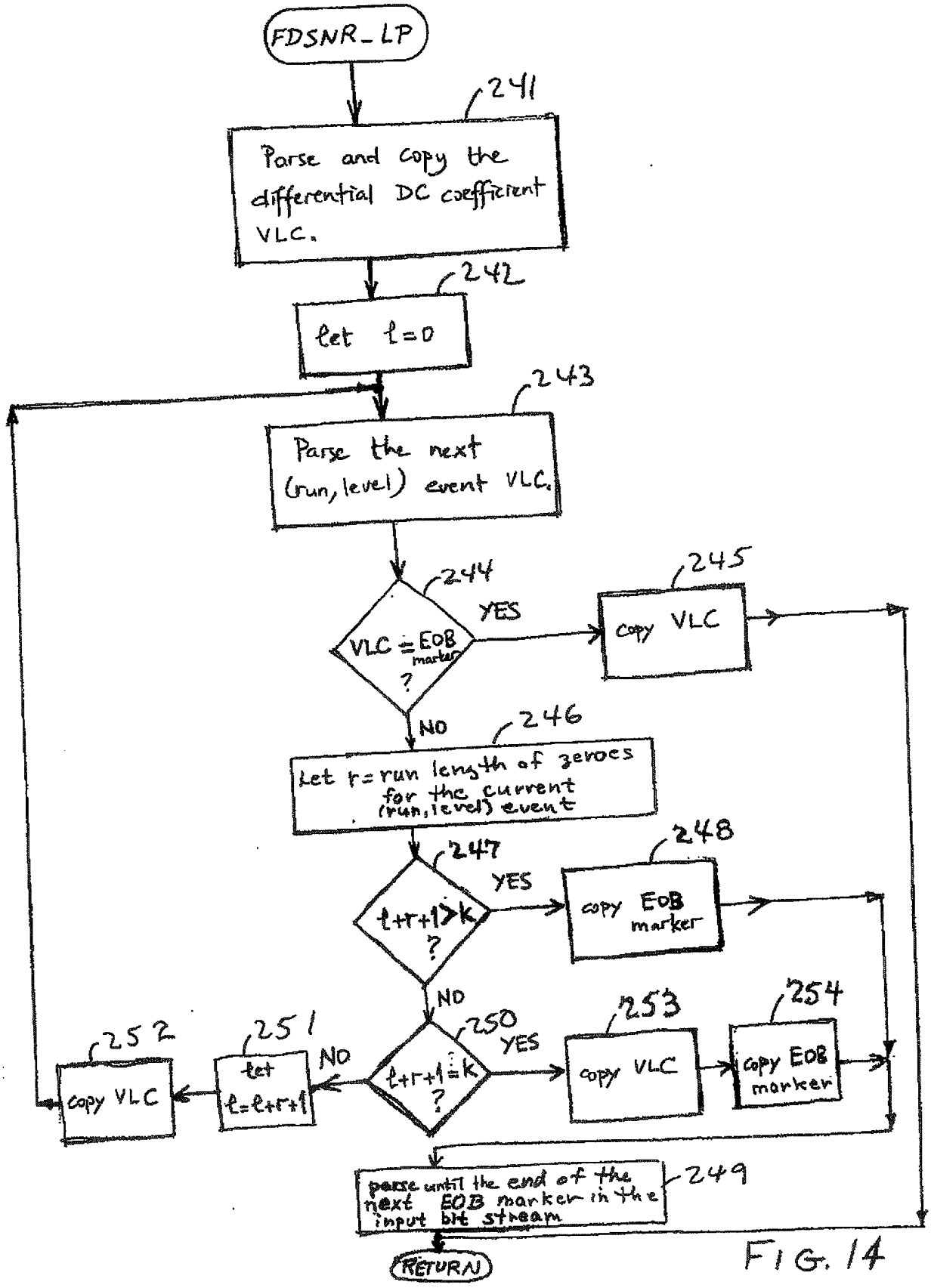
FDSNR_LP
241
Parse and copy the differential DC coefficient VLC.
242
let t=0
243
Parse the next (run, level) event VLC.
244
VLC = EOB marker ?
YES
245
copy VLC
NO
246
Let r = run length of zeroes for the current (run, level) event
247
t+r+1>k ?
YES
248
copy EOB marker
NO
250
t+r+1=k ?
YES
253
copy VLC
254
copy EOB marker
NO
251
let t=t+r+1
252
copy VLC
249
parse until the end of the next EOB marker in the input bit stream
RETURN

FIG. 14

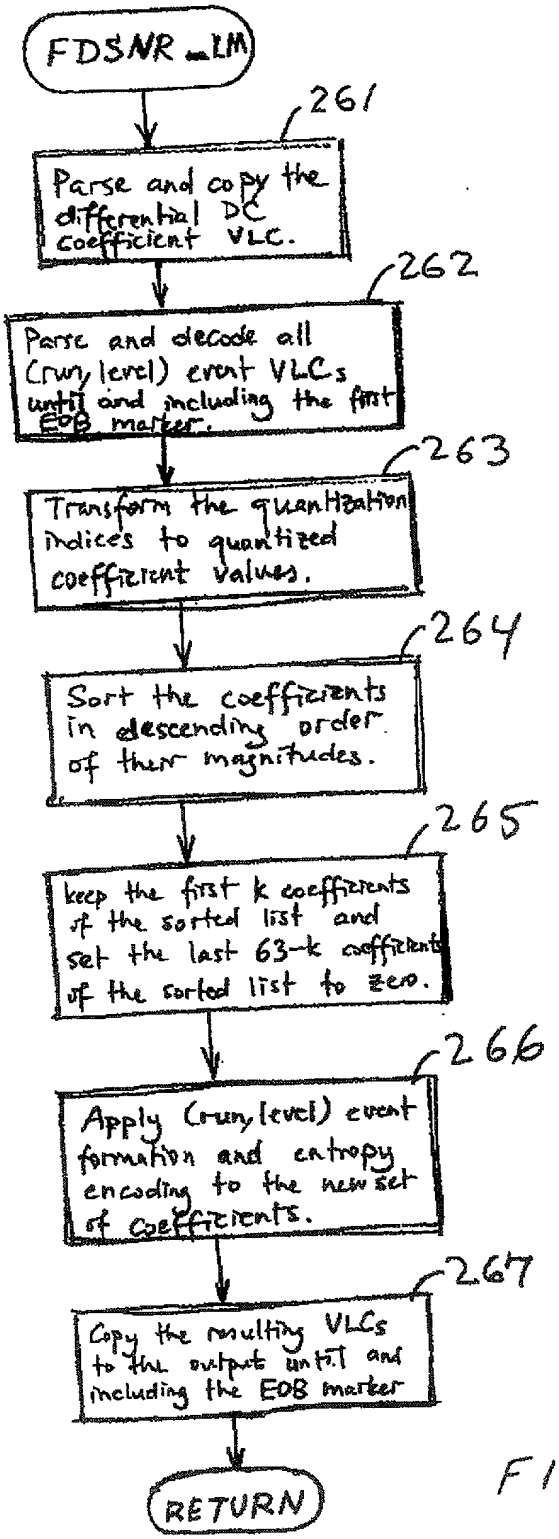
FDSNR_LM
261
Parse and copy the differential DC coefficient VLC.
262
Parse and decode all (run, level) event VLCs until and including the first EOB marker.
263
Transform the quantization indices to quantized coefficient values.
264
Sort the coefficients in descending order of their magnitudes.
265
keep the first k coefficients of the sorted list and set the last 63-k coefficients of the sorted list to zero.
266
Apply (run, level) event formation and entropy encoding to the new set of coefficients.
267
Copy the resulting VLCs to the output until and including the EOB marker
RETURN

FIG. 15

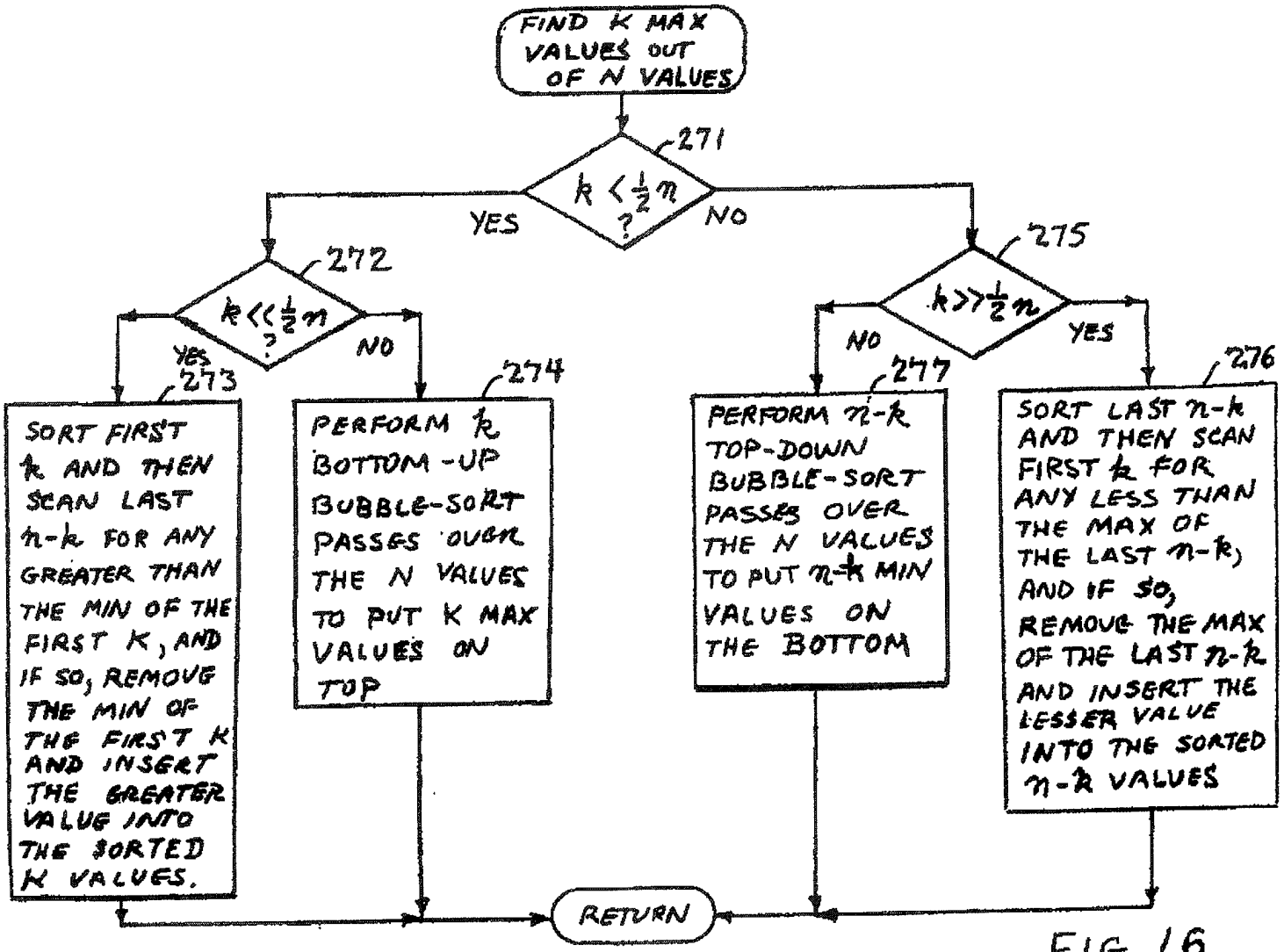
FIND K MAX VALUES OUT OF N VALUES
271
k < ½n ?
YES
NO
272
k << ½n ?
YES
NO
273
SORT FIRST k AND THEN SCAN LAST n-k FOR ANY GREATER THAN THE MIN OF THE FIRST K, AND IF SO, REMOVE THE MIN OF THE FIRST K AND INSERT THE GREATER VALUE INTO THE SORTED K VALUES.
274
PERFORM k BOTTOM-UP BUBBLE-SORT PASSES OVER THE N VALUES TO PUT K MAX VALUES ON TOP
275
k >> ½n ?
NO
YES
277
PERFORM n-k TOP-DOWN BUBBLE-SORT PASSES OVER THE N VALUES TO PUT n-k MIN VALUES ON THE BOTTOM
276
SORT LAST n-k AND THEN SCAN FIRST k FOR ANY LESS THAN THE MAX OF THE LAST n-k) AND IF SO, REMOVE THE MAX OF THE LAST n-k AND INSERT THE LESSER VALUE INTO THE SORTED n-k VALUES
RETURN

FIG. 16

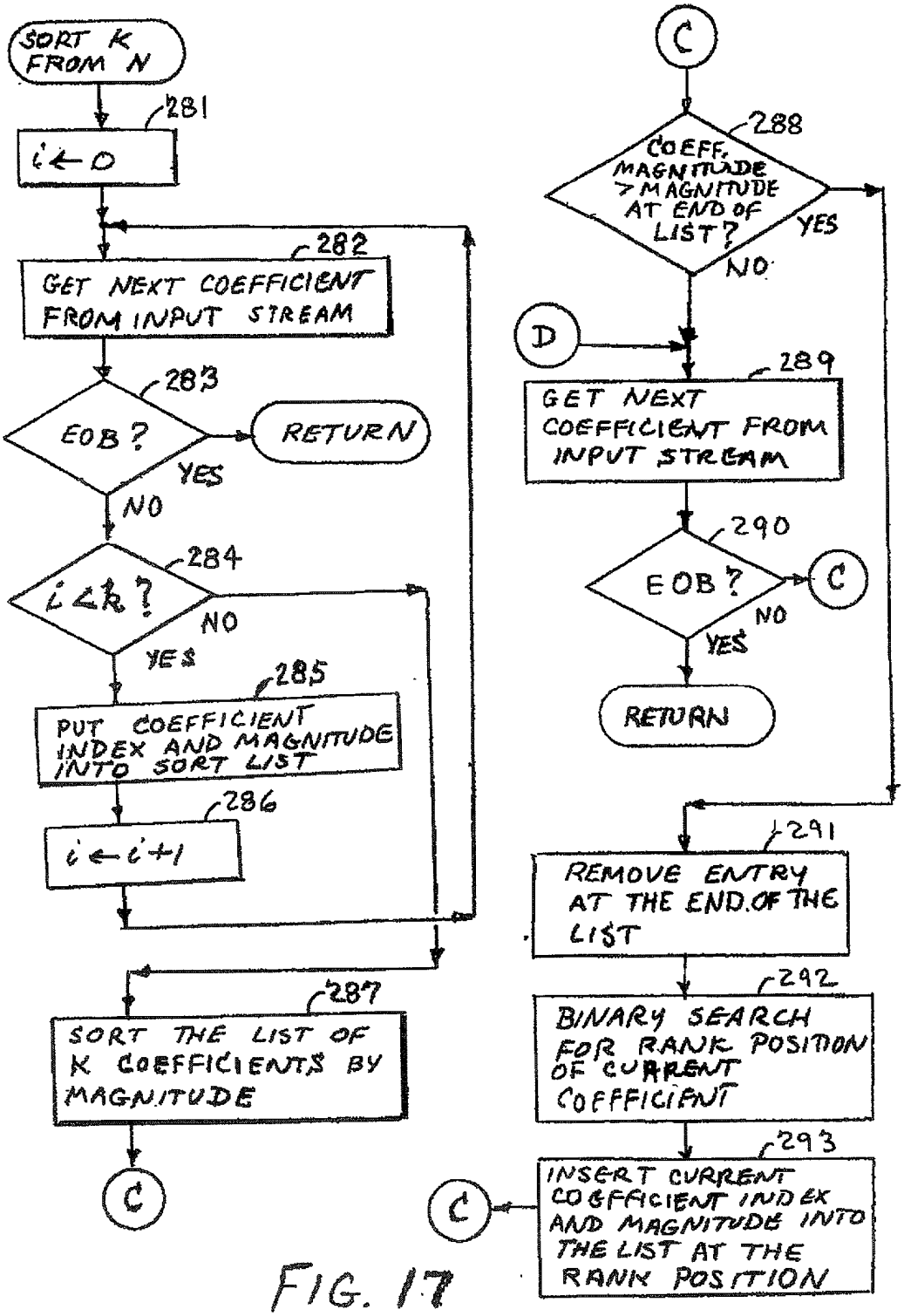
SORT K FROM N
281
i ← 0
282
GET NEXT COEFFICIENT FROM INPUT STREAM
283
EOB ?
YES
RETURN
NO
284
i < k ?
NO
YES
285
PUT COEFFICIENT INDEX AND MAGNITUDE INTO SORT LIST
286
i ← i+1
287
SORT THE LIST OF K COEFFICIENTS BY MAGNITUDE
C
C
288
COEFF. MAGNITUDE > MAGNITUDE AT END OF LIST ?
YES
NO
D
289
GET NEXT COEFFICIENT FROM INPUT STREAM
290
EOB ?
C
NO
YES
RETURN
291
REMOVE ENTRY AT THE END OF THE LIST
292
BINARY SEARCH FOR RANK POSITION OF CURRENT COFFFICIENT
293
INSERT CURRENT COEFFICIENT INDEX AND MAGNITUDE INTO THE LIST AT THE RANK POSITION
C

FIG. 17

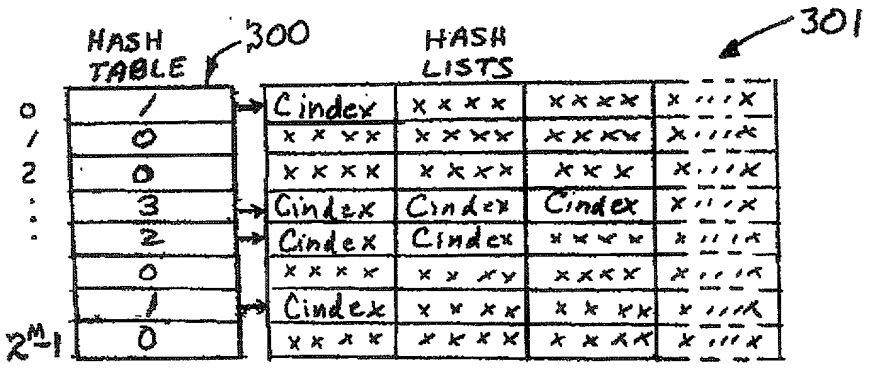
HASH TABLE
300
HASH LISTS
301
0
1
2
2^M-1
1
0
0
3
2
0
1
0
Cindex
Cindex
Cindex
Cindex
Cindex
Cindex

FIG. 1B

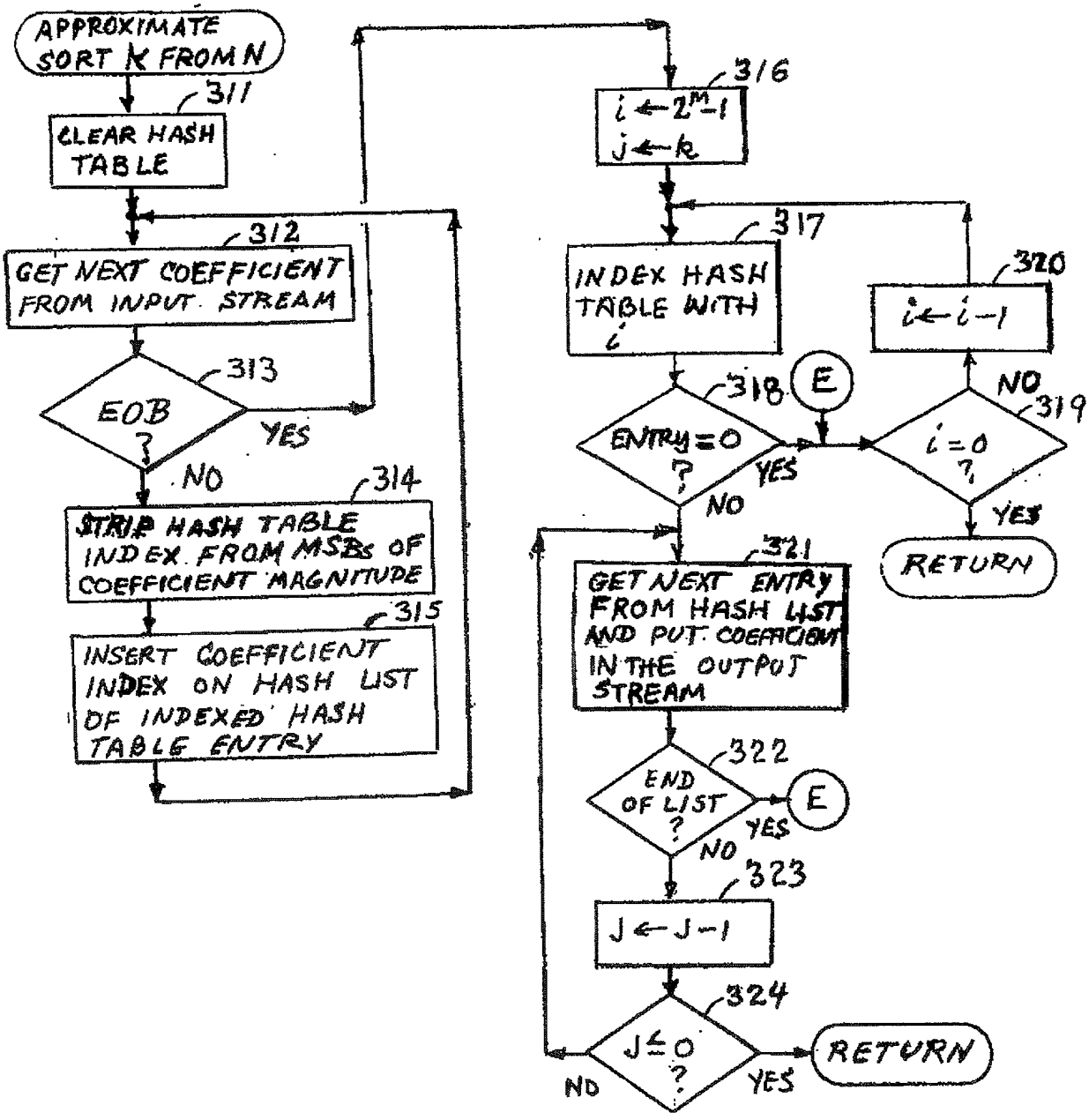
APPROXIMATE SORT K FROM N
311
CLEAR HASH TABLE
312
GET NEXT COEFFICIENT FROM INPUT STREAM
313
EOB ?
YES
NO
314
STRIP HASH TABLE INDEX FROM MSBs OF COEFFICIENT MAGNITUDE
315
INSERT COEFFICIENT INDEX ON HASH LIST OF INDEXED HASH TABLE ENTRY
316
i ← 2^M-1
j ← k
317
INDEX HASH TABLE WITH i
318
ENTRY=0 ?
E
YES
NO
319
i=0 ?
NO
YES
320
i ← i-1
RETURN
321
GET NEXT ENTRY FROM HASH LIST AND PUT COEFFICIENT IN THE OUTPUT STREAM
322
END OF LIST ?
E
YES
NO
323
J ← J-1
324
J≤0 ?
NO
YES
RETURN

FIG. 19

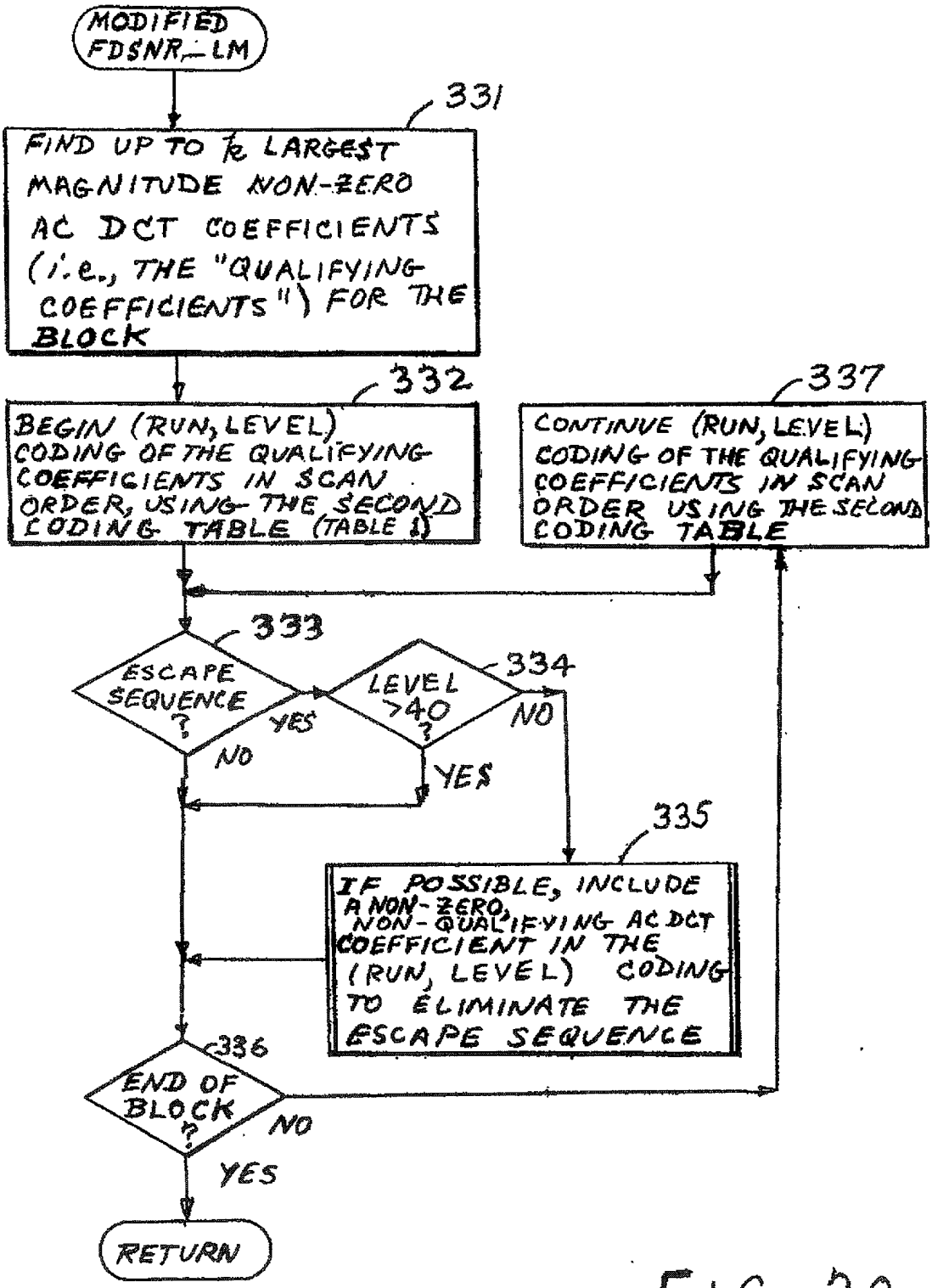
MODIFIED FDSNR_LM
331
FIND UP TO k LARGEST MAGNITUDE NON-ZERO AC DCT COEFFICIENTS (i.e., THE "QUALIFYING COEFFICIENTS") FOR THE BLOCK
332
BEGIN (RUN, LEVEL) CODING OF THE QUALIFYING COEFFICIENTS IN SCAN ORDER, USING THE SECOND CODING TABLE (TABLE 1)
337
CONTINUE (RUN, LEVEL) CODING OF THE QUALIFYING COEFFICIENTS IN SCAN ORDER USING THE SECOND CODING TABLE
333
ESCAPE SEQUENCE ?
YES
NO
334
LEVEL >40 ?
NO
YES
335
IF POSSIBLE, INCLUDE A NON-ZERO, NON-QUALIFYING AC DCT COEFFICIENT IN THE (RUN, LEVEL) CODING TO ELIMINATE THE ESCAPE SEQUENCE
336
END OF BLOCK ?
NO
YES
RETURN

FIG. 20

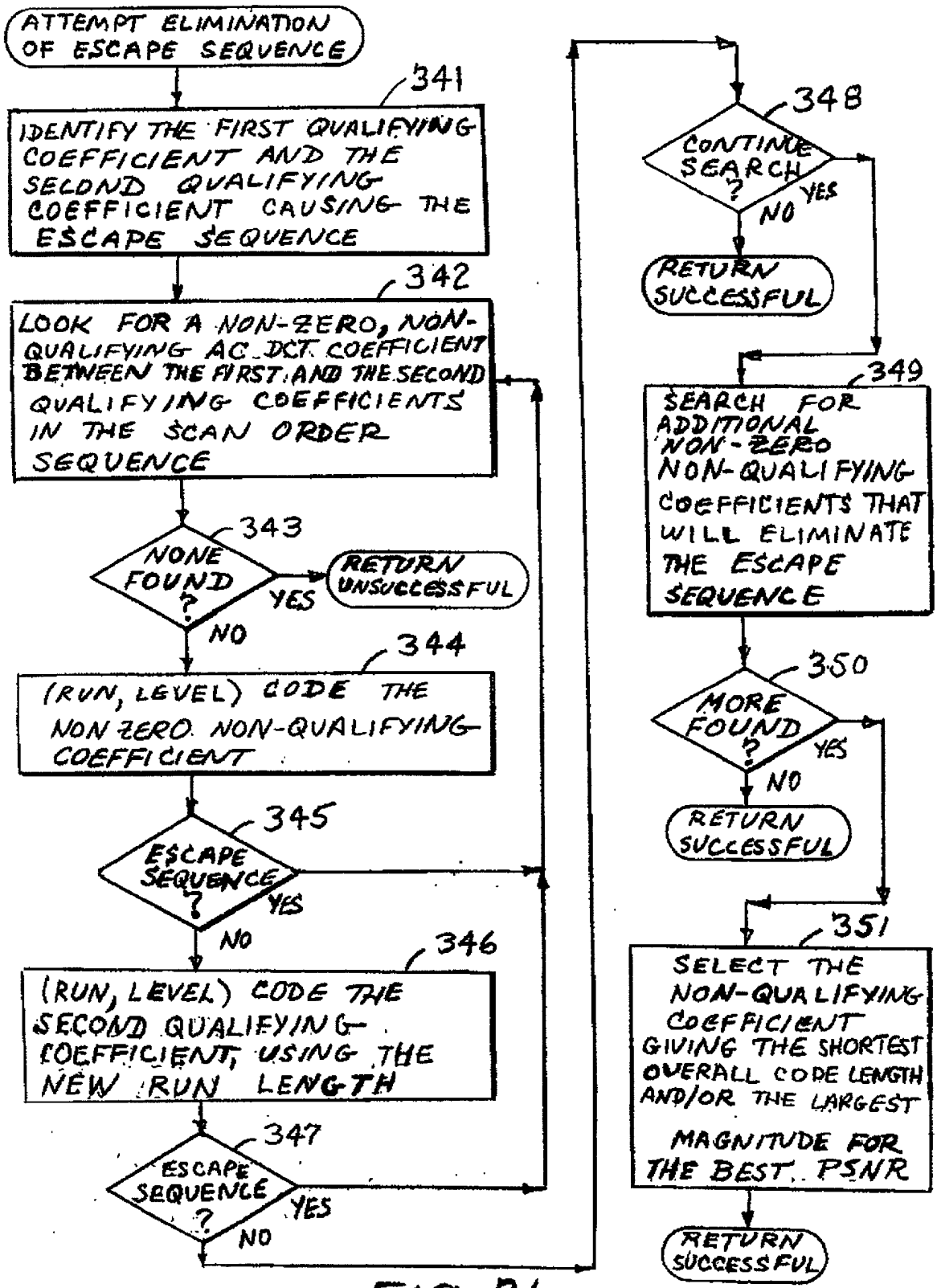
ATTEMPT ELIMINATION OF ESCAPE SEQUENCE
341
IDENTIFY THE FIRST QUALIFYING COEFFICIENT AND THE SECOND QUALIFYING COEFFICIENT CAUSING THE ESCAPE SEQUENCE
342
LOOK FOR A NON-ZERO, NON-QUALIFYING AC DCT COEFFICIENT BETWEEN THE FIRST AND THE SECOND QUALIFYING COEFFICIENTS IN THE SCAN ORDER SEQUENCE
343
NONE FOUND ?
YES
RETURN UNSUCCESSFUL
NO
344
(RUN, LEVEL) CODE THE NON ZERO NON-QUALIFYING COEFFICIENT
345
ESCAPE SEQUENCE ?
YES
NO
346
(RUN, LEVEL) CODE THE SECOND QUALIFYING COEFFICIENT, USING THE NEW RUN LENGTH
347
ESCAPE SEQUENCE ?
YES
NO
348
CONTINUE SEARCH ?
YES
NO
RETURN SUCCESSFUL
349
SEARCH FOR ADDITIONAL NON-ZERO NON-QUALIFYING COEFFICIENTS THAT WILL ELIMINATE THE ESCAPE SEQUENCE
350
MORE FOUND ?
YES
NO
RETURN SUCCESSFUL
351
SELECT THE NON-QUALIFYING COEFFICIENT GIVING THE SHORTEST OVERALL CODE LENGTH AND/OR THE LARGEST MAGNITUDE FOR THE BEST PSNR
RETURN SUCCESSFUL

FIG. 21

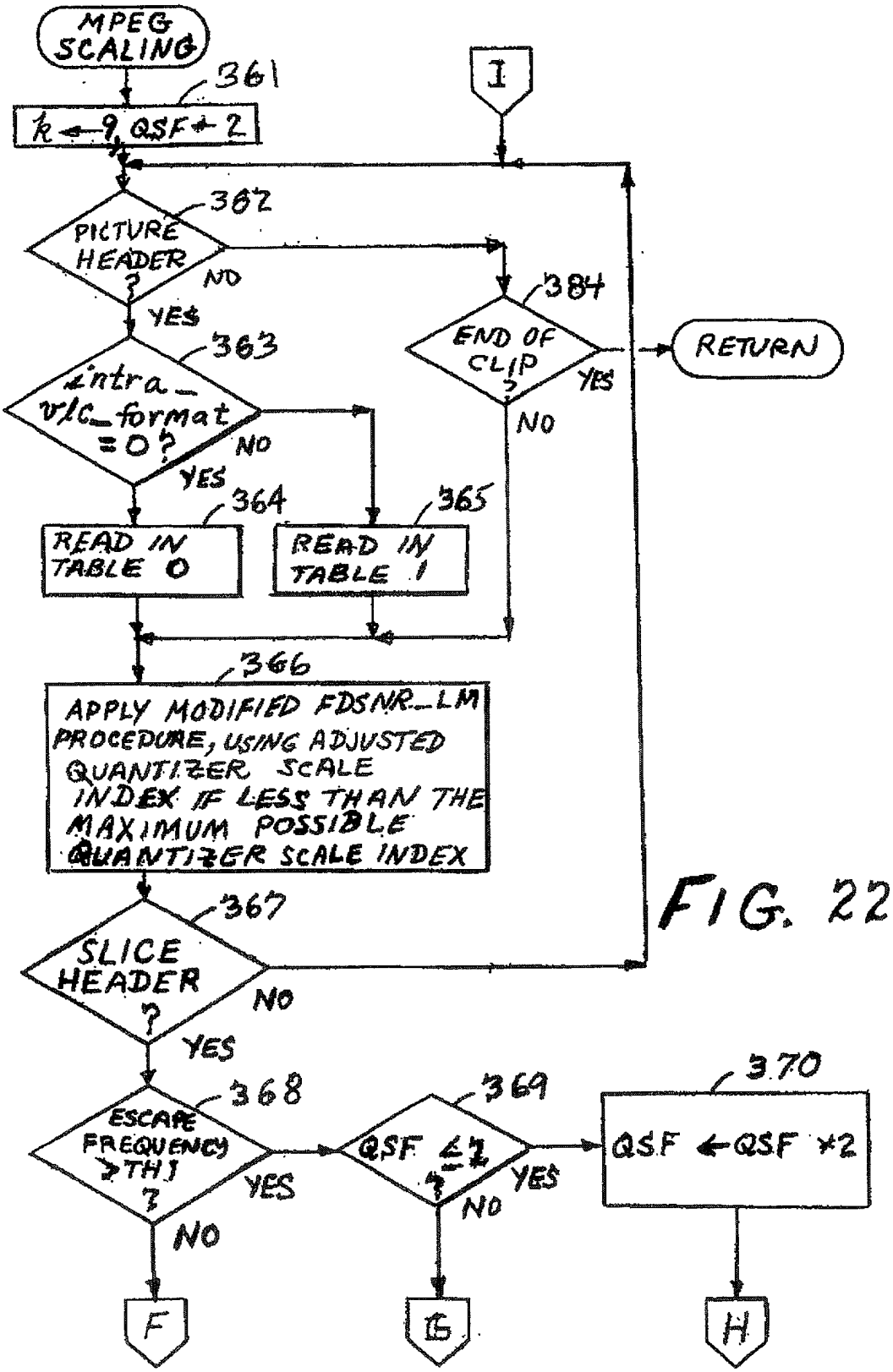

FIG. 22

5,956,554 - 12/1999

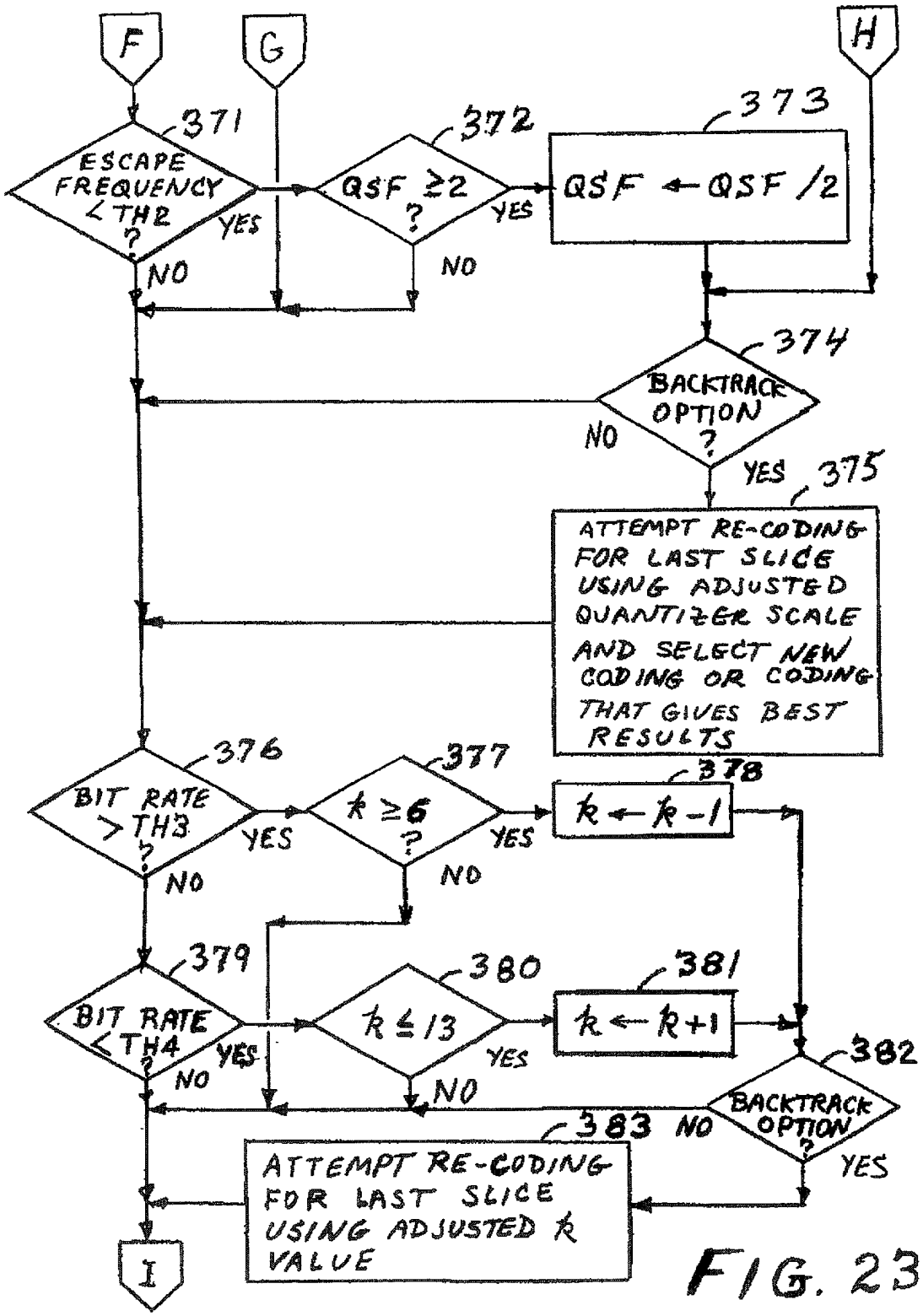
F
G
H
371
ESCAPE FREQUENCY < TH2 ?
YES
NO
372
QSF ≥ 2 ?
YES
NO
373
QSF ← QSF / 2
374
BACKTRACK OPTION ?
NO
YES
375
ATTEMPT RE-CODING FOR LAST SLICE USING ADJUSTED QUANTIZER SCALE AND SELECT NEW CODING OR CODING THAT GIVES BEST RESULTS
376
BIT RATE > TH3 ?
YES
NO
377
k ≥ 6 ?
YES
NO
378
k ← k - 1
379
BIT RATE < TH4 ?
YES
NO
380
k ≤ 13
YES
NO
381
k ← k + 1
382
BACKTRACK OPTION ?
NO
YES
383
ATTEMPT RE-CODING FOR LAST SLICE USING ADJUSTED k VALUE
I

FIG. 23

00827P-US09260

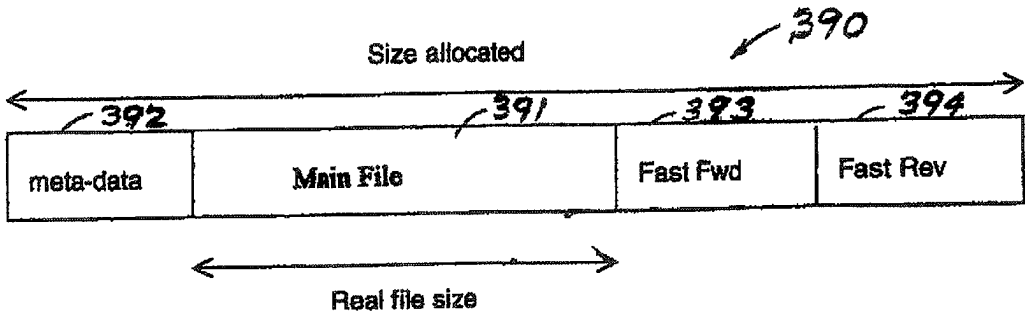

FIG. 24

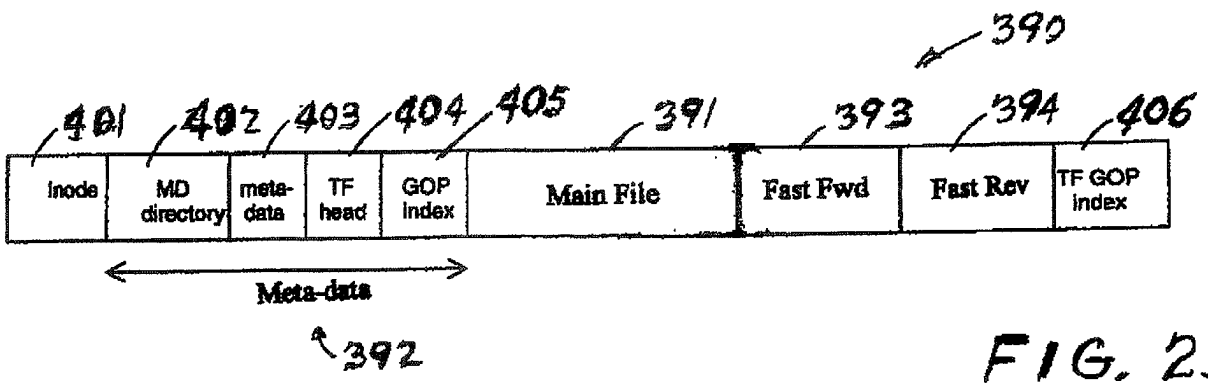

FIG. 25

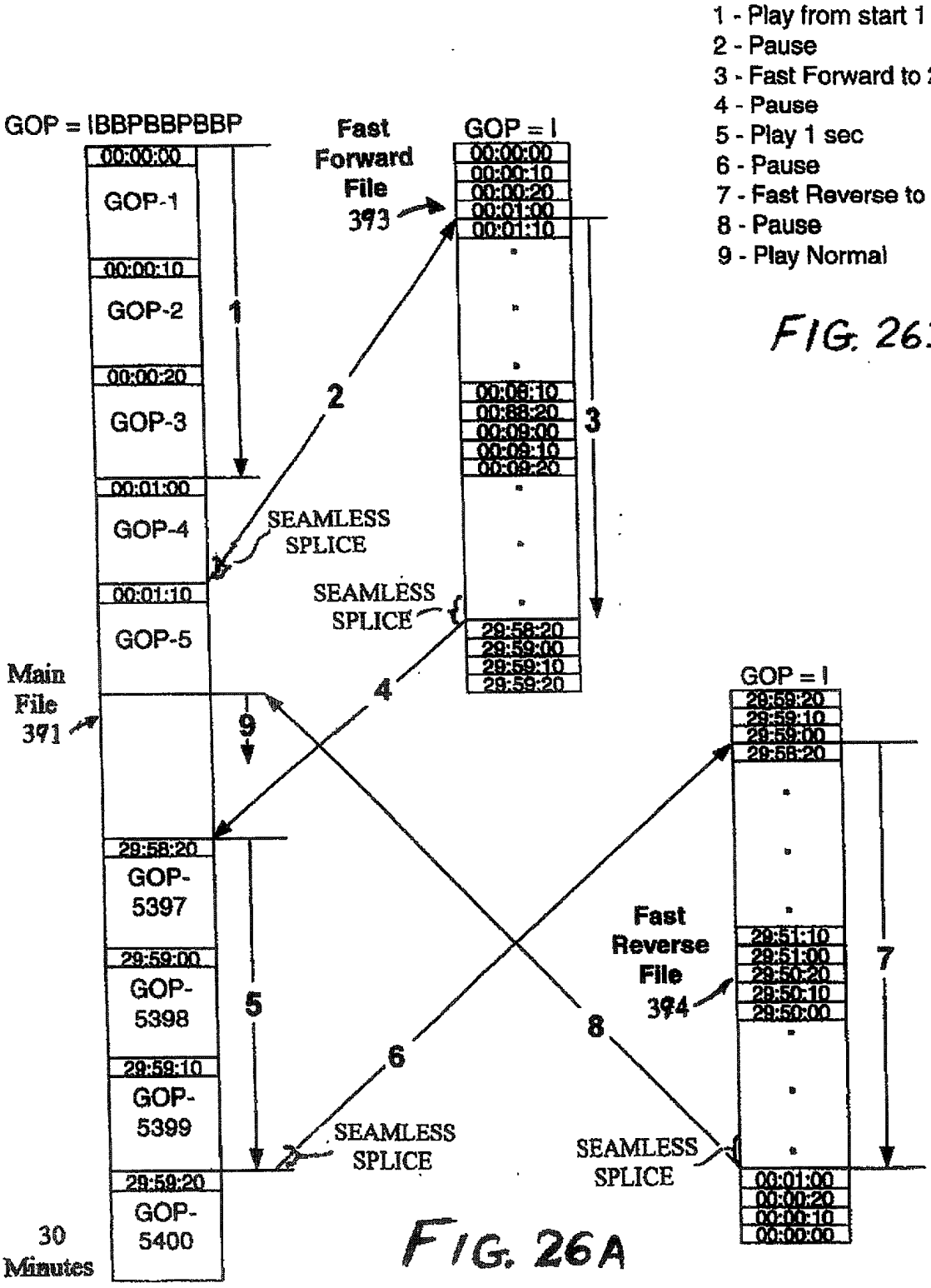

FIG. 26A

1 - Play from start 1 sec
2 - Pause
3 - Fast Forward to 29 min
4 - Pause
5 - Play 1 sec
6 - Pause
7 - Fast Reverse to 1 sec
8 - Pause
9 - Play Normal

FIG. 26B

	READ	WRITE
Copy of the asset with all the data	EMPEG2	EMPEG2
Copy only the main asset	RAW	MPEG2
Archive	EMPEG2	EMPEG2
Play	MPEG2	
Record		MPEG2

FIG. 27

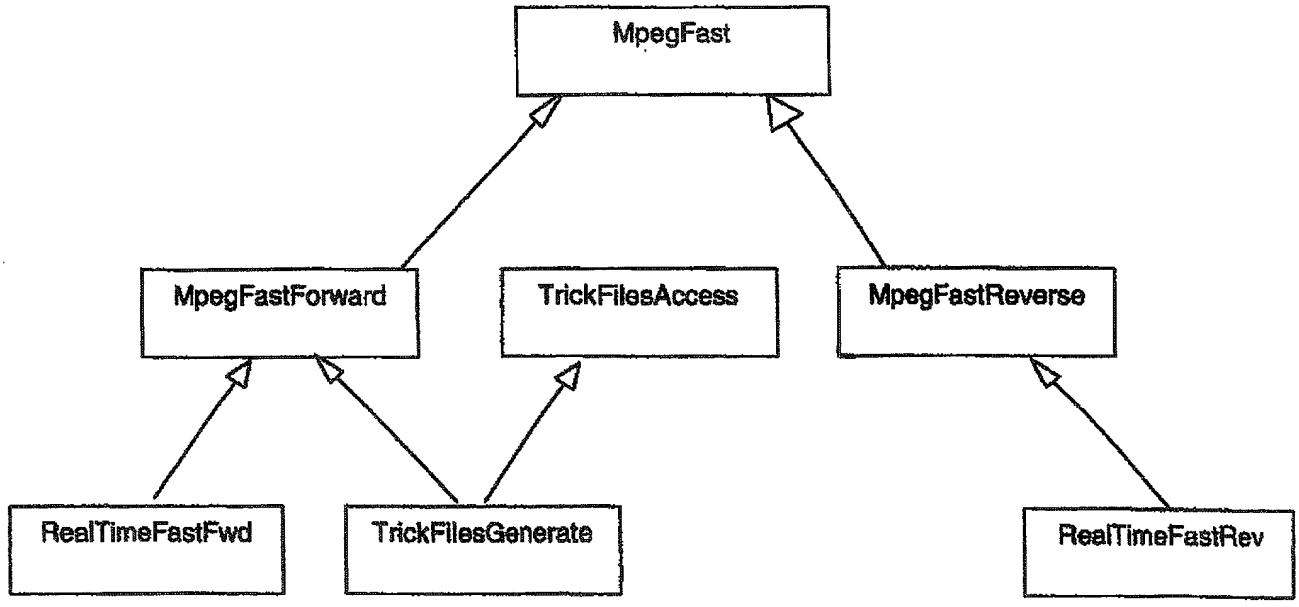

FIG. 28

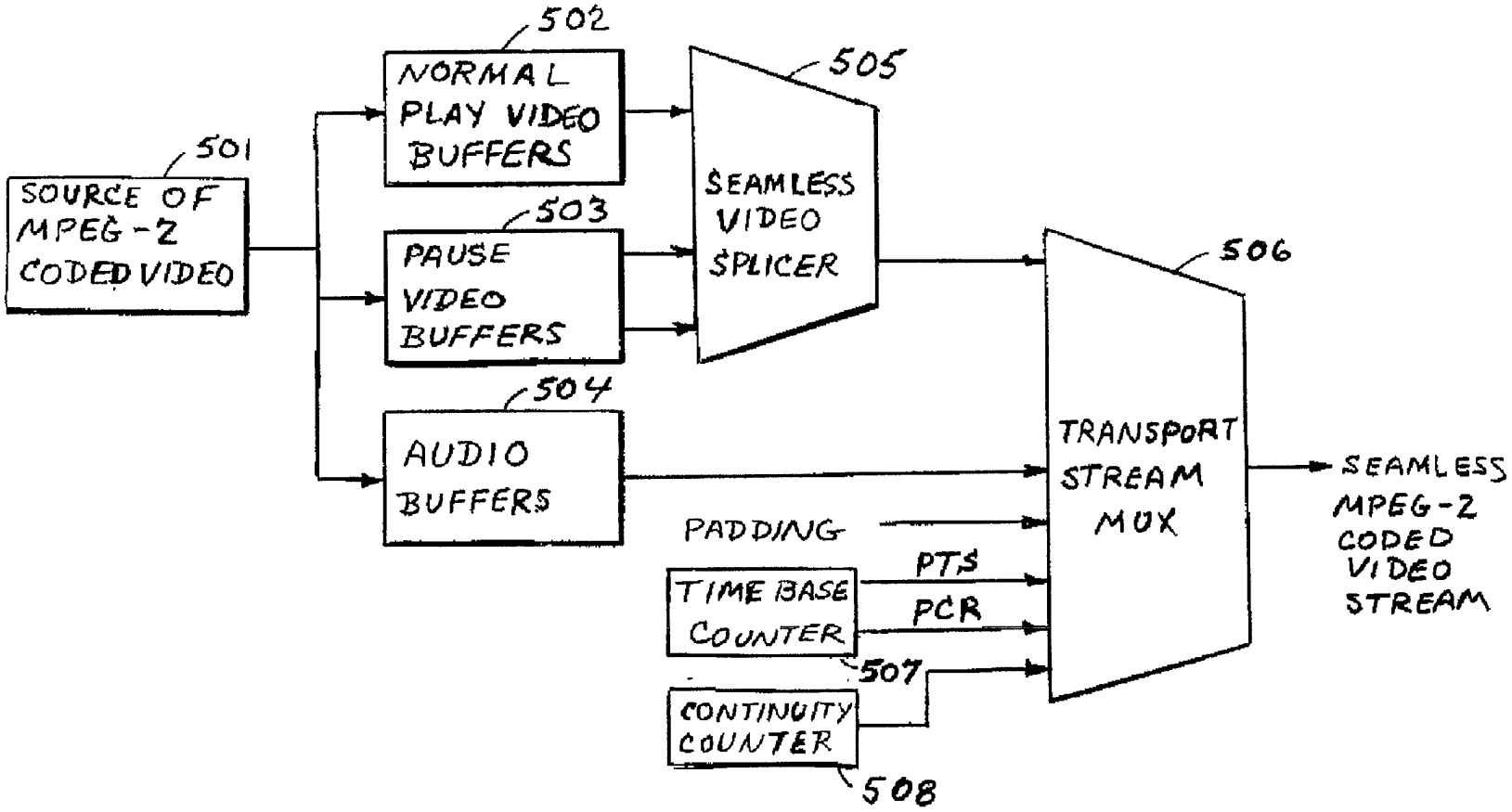
502
NORMAL PLAY VIDEO BUFFERS
505
501
SOURCE OF MPEG-2 CODED VIDEO
503
PAUSE VIDEO BUFFERS
SEAMLESS VIDEO SPLICER
506
504
AUDIO BUFFERS
TRANSPORT STREAM MUX
SEAMLESS MPEG-2 CODED VIDEO STREAM
PADDING
PTS
PCR
TIME BASE COUNTER
507
CONTINUITY COUNTER
508

FIG. 29

ORIGINAL VIDEO ES

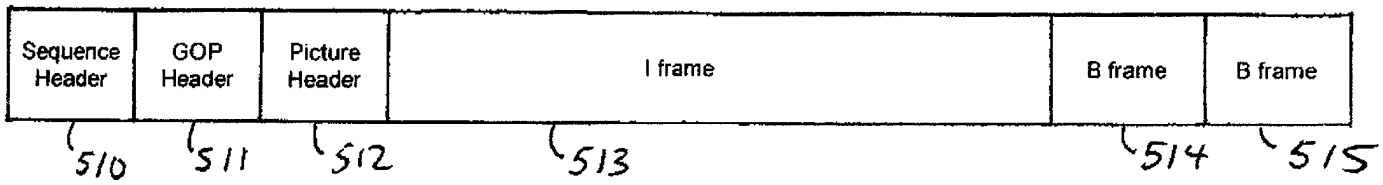

FIG. 30

P FREEZE FRAME ES

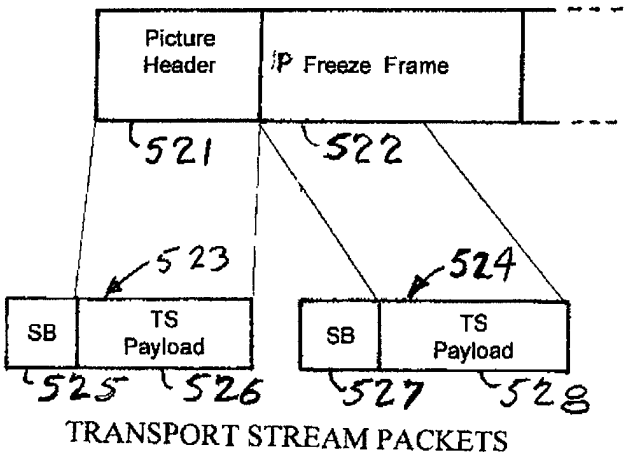

FIG. 31

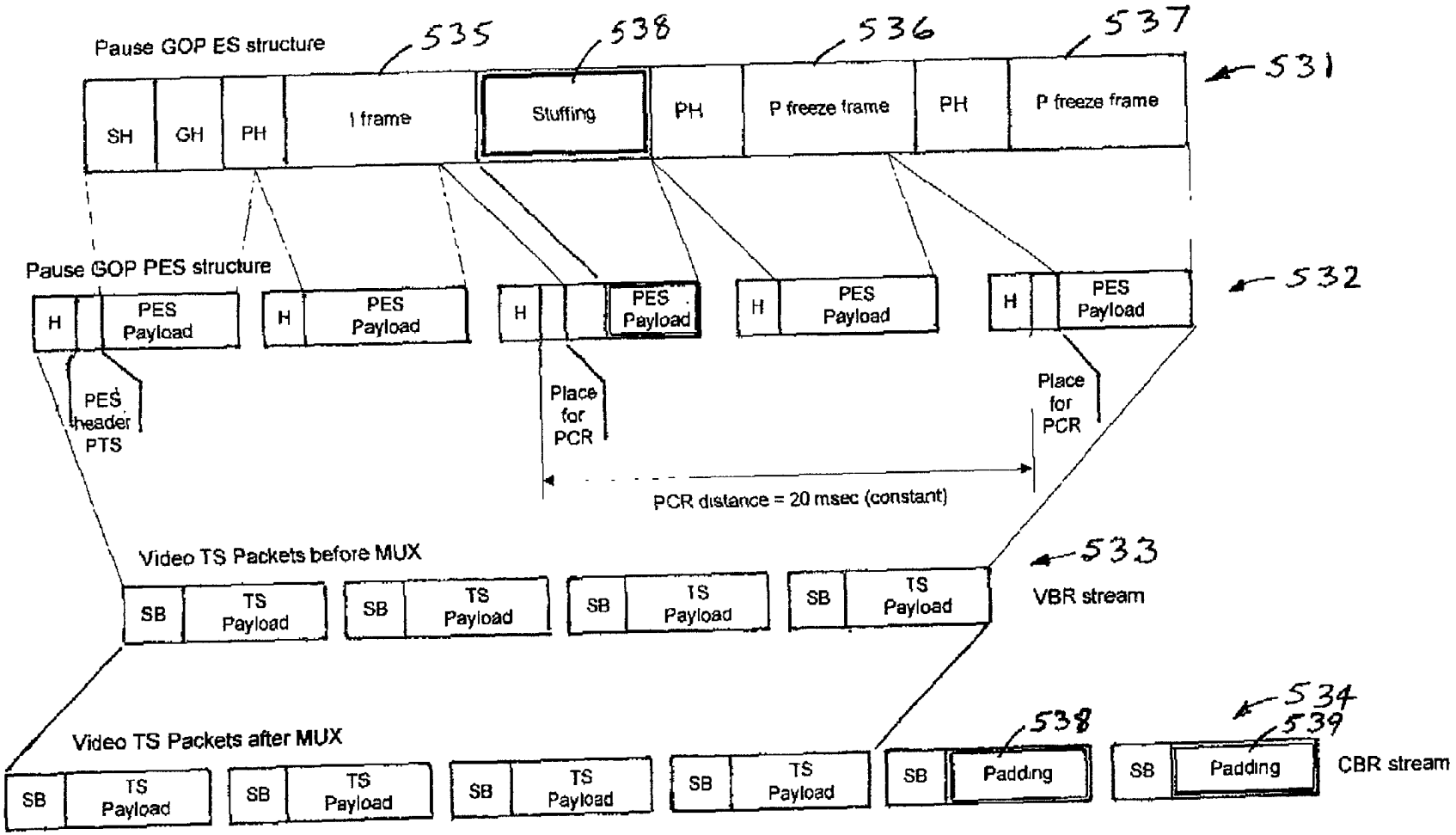
Pause GOP ES structure
535
538
536
537
531
SH
GH
PH
I frame
Stuffing
PH
P freeze frame
PH
P freeze frame
Pause GOP PES structure
532
H
PES Payload
PES header PTS
Place for PCR
PCR distance = 20 msec (constant)
Video TS Packets before MUX
533
VBR stream
SB
TS Payload
Video TS Packets after MUX
534
538
539
Padding
CBR stream

FIG. 32

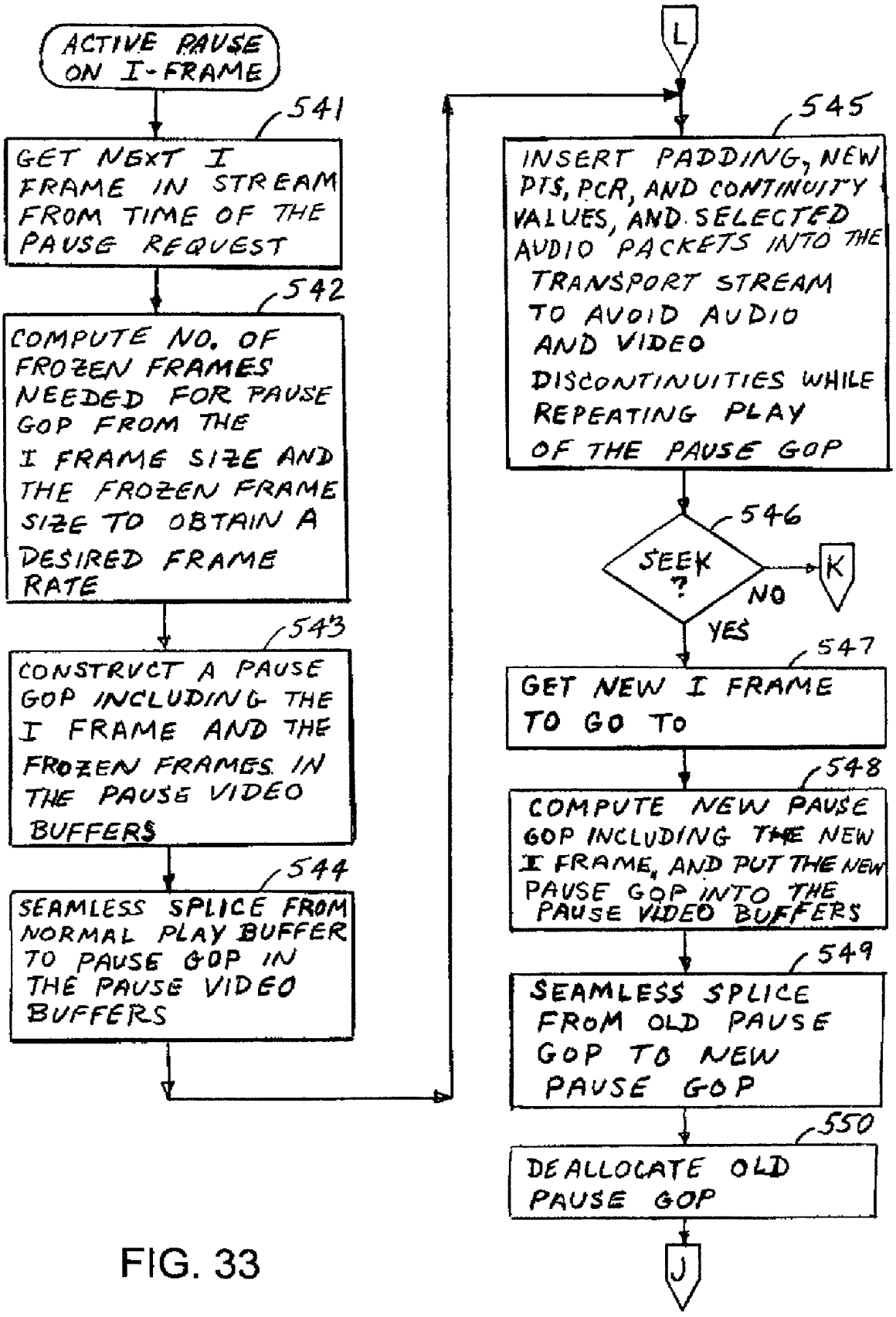
ACTIVE PAUSE ON I-FRAME
541
GET NEXT I FRAME IN STREAM FROM TIME OF THE PAUSE REQUEST
542
COMPUTE NO. OF FROZEN FRAMES NEEDED FOR PAUSE GOP FROM THE I FRAME SIZE AND THE FROZEN FRAME SIZE TO OBTAIN A DESIRED FRAME RATE
543
CONSTRUCT A PAUSE GOP INCLUDING THE I FRAME AND THE FROZEN FRAMES IN THE PAUSE VIDEO BUFFERS
544
SEAMLESS SPLICE FROM NORMAL PLAY BUFFER TO PAUSE GOP IN THE PAUSE VIDEO BUFFERS
L
545
INSERT PADDING, NEW PTS, PCR, AND CONTINUITY VALUES, AND SELECTED AUDIO PACKETS INTO THE TRANSPORT STREAM TO AVOID AUDIO AND VIDEO DISCONTINUITIES WHILE REPEATING PLAY OF THE PAUSE GOP
546
SEEK ?
NO
K
YES
547
GET NEW I FRAME TO GO TO
548
COMPUTE NEW PAUSE GOP INCLUDING THE NEW I FRAME, AND PUT THE NEW PAUSE GOP INTO THE PAUSE VIDEO BUFFERS
549
SEAMLESS SPLICE FROM OLD PAUSE GOP TO NEW PAUSE GOP
550
DEALLOCATE OLD PAUSE GOP
J

FIG. 33

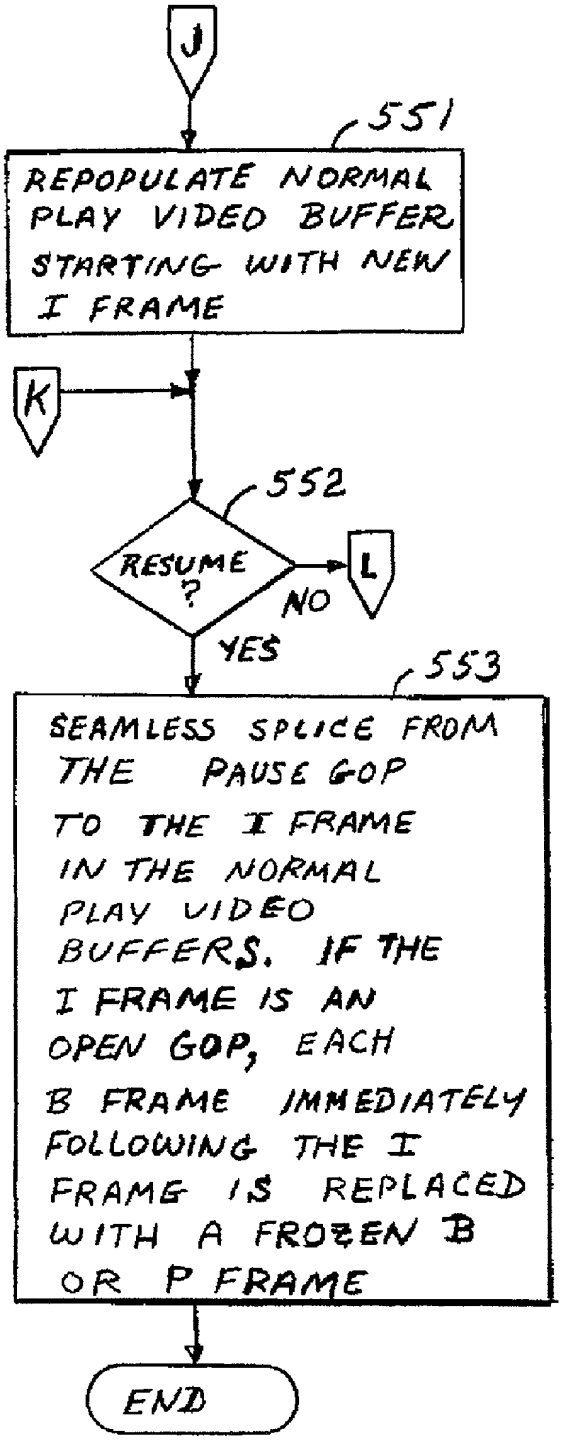
J
551
REPOPULATE NORMAL PLAY VIDEO BUFFER STARTING WITH NEW I FRAME
K
552
RESUME ?
NO
L
YES
553
SEAMLESS SPLICE FROM THE PAUSE GOP TO THE I FRAME IN THE NORMAL PLAY VIDEO BUFFERS. IF THE I FRAME IS AN OPEN GOP, EACH B FRAME IMMEDIATELY FOLLOWING THE I FRAME IS REPLACED WITH A FROZEN B OR P FRAME
END

FIG. 34

DOCKET NO. 50

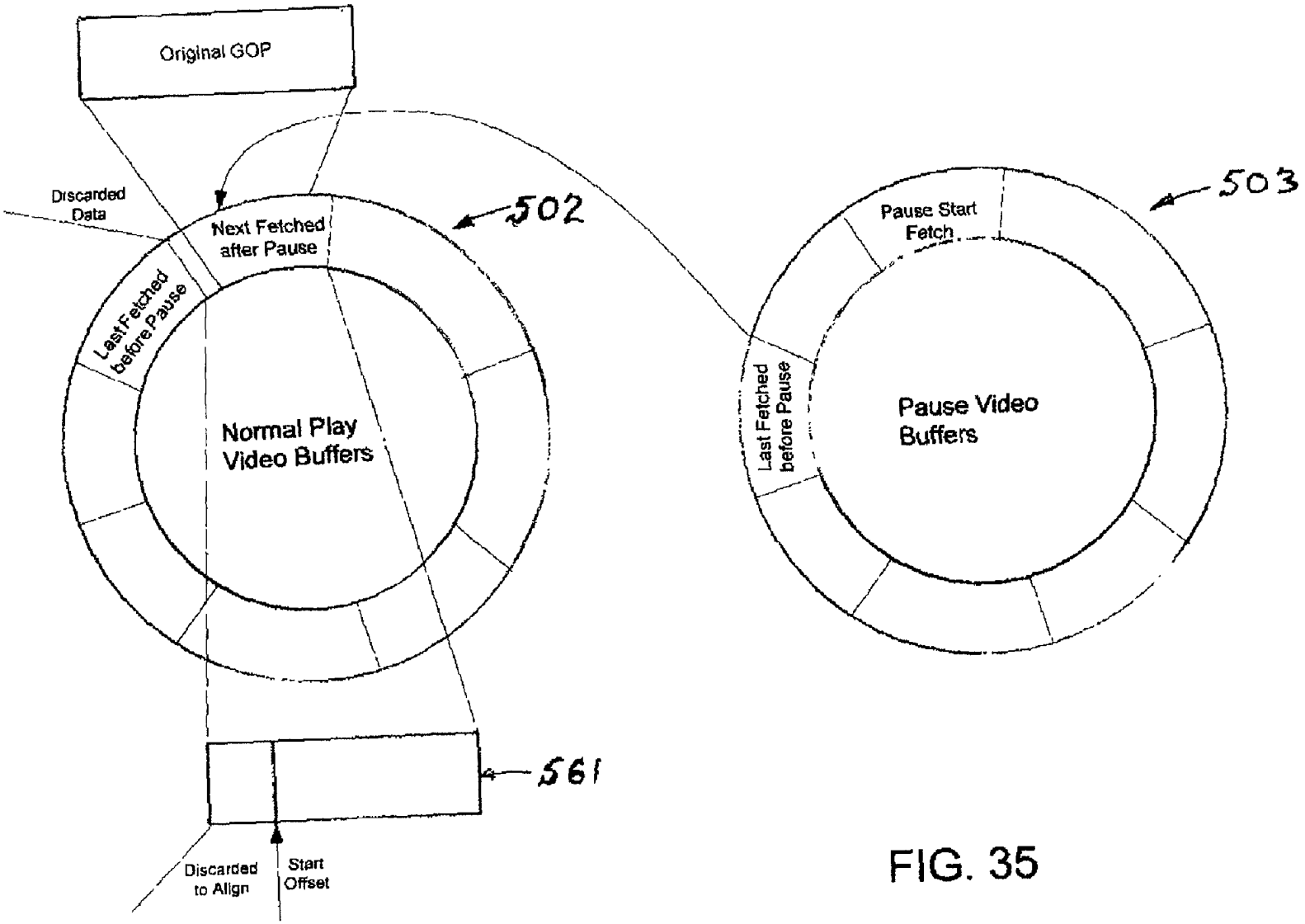

FIG. 35

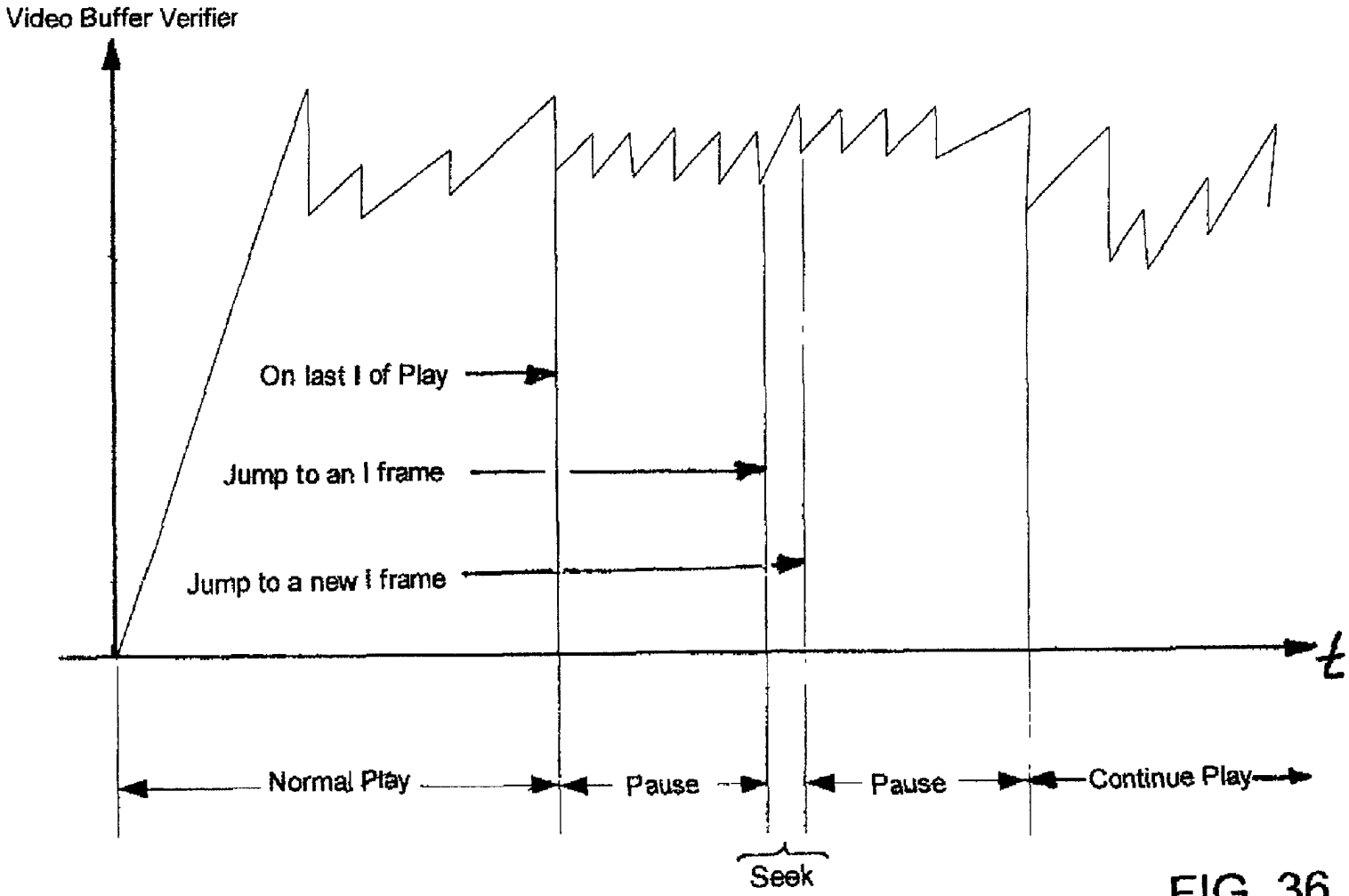
Video Buffer Verifier
On last I of Play
Jump to an I frame
Jump to a new I frame
t
Normal Play
Pause
Pause
Continue Play
Seek

FIG. 36

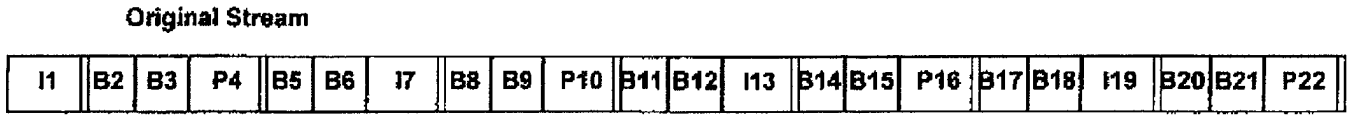

FIG. 37

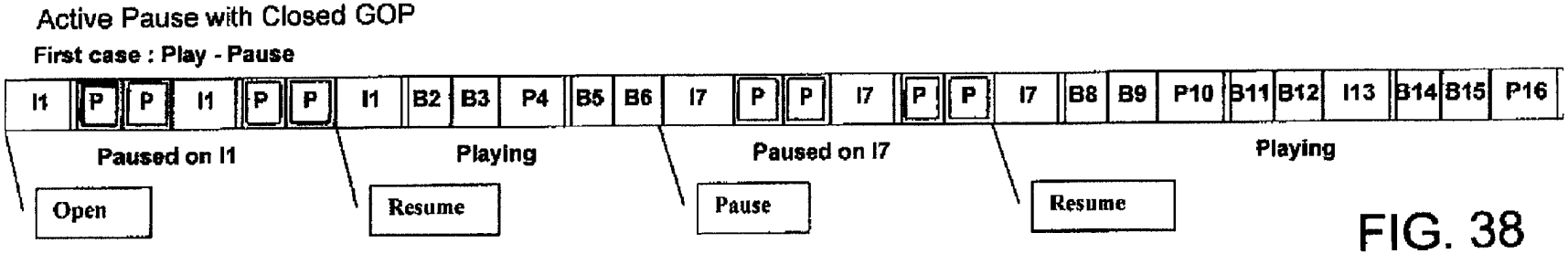

FIG. 38

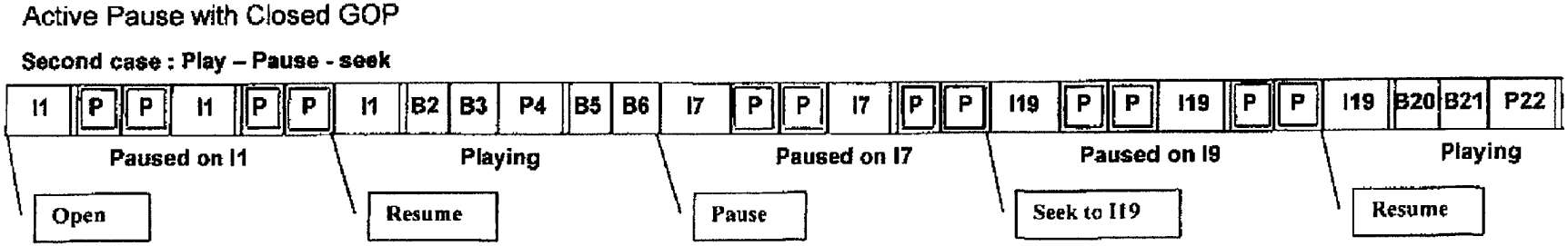

FIG. 39

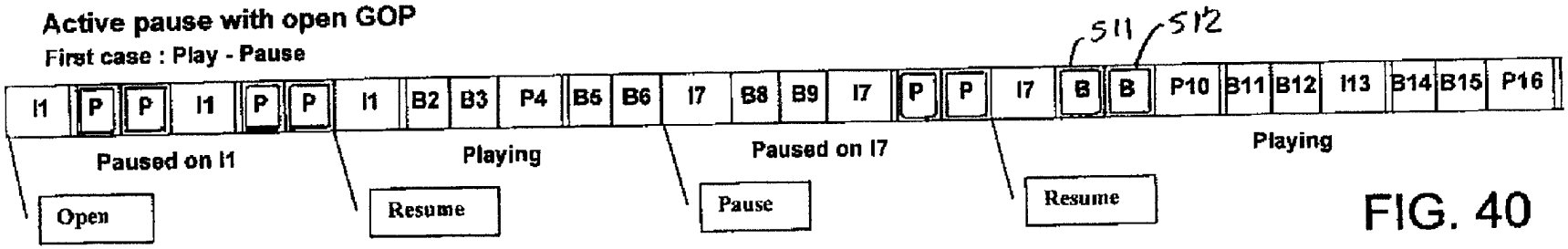
Active pause with open GOP
First case : Play - Pause
511
512
I1 P P I1 P P I1 B2 B3 P4 B5 B6 I7 B8 B9 I7 P P I7 B B P10 B11 B12 I13 B14 B15 P16
Paused on I1
Playing
Paused on I7
Playing
Open
Resume
Pause
Resume

FIG. 40

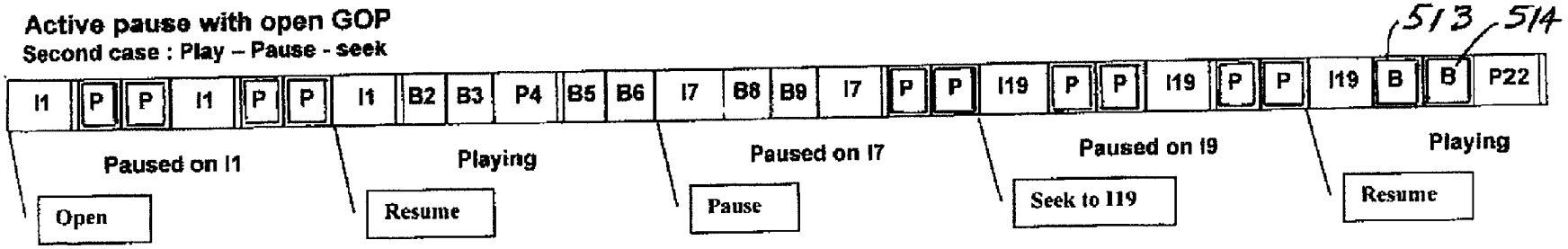
Active pause with open GOP
Second case : Play – Pause - seek
513
514
I1 P P I1 P P I1 B2 B3 P4 B5 B6 I7 B8 B9 I7 P P I19 P P I19 P P I19 B B P22
Paused on I1
Playing
Paused on I7
Paused on I9
Playing
Open
Resume
Pause
Seek to I19
Resume

FIG. 41

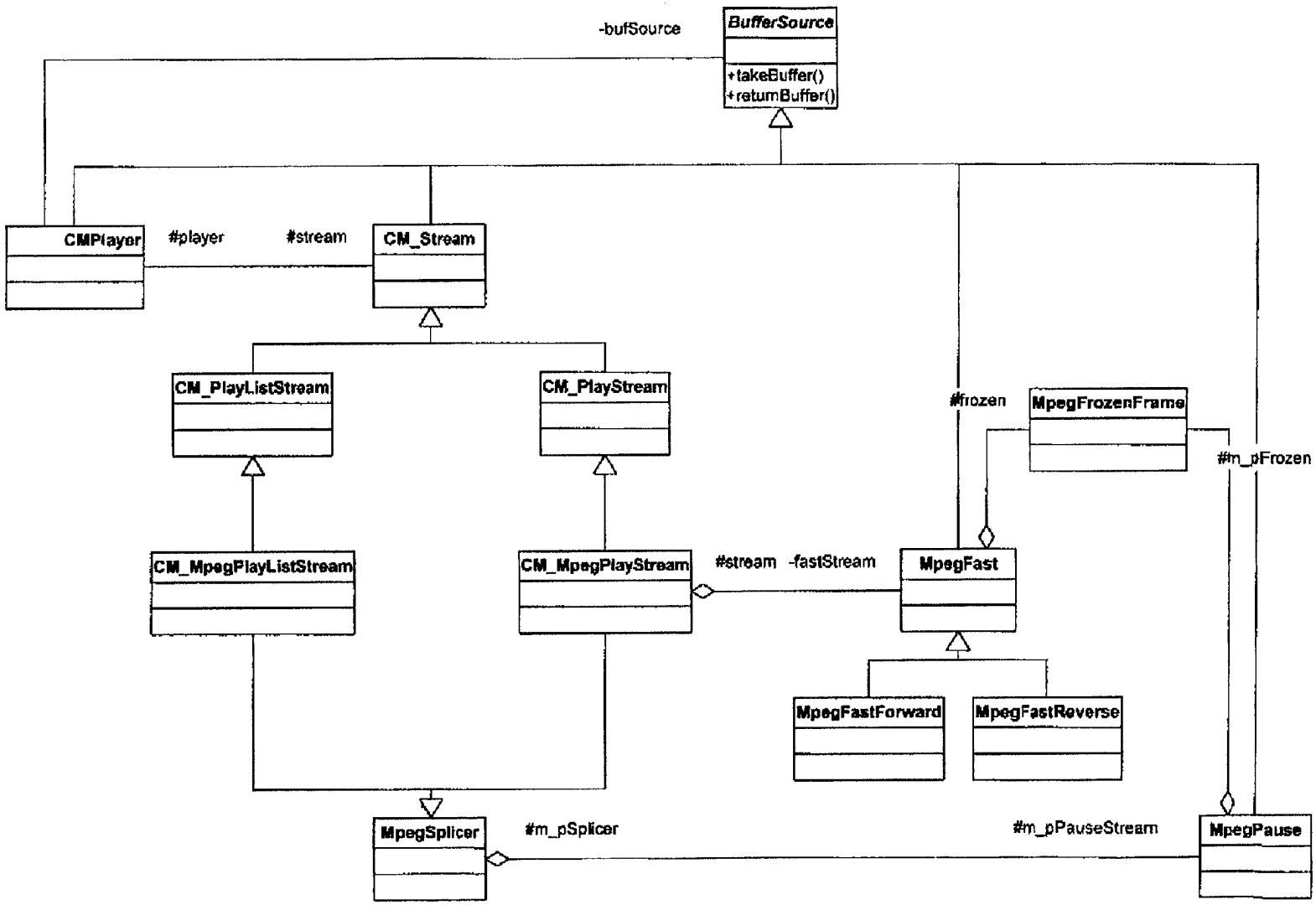
-bufSource
BufferSource
+takeBuffer()
+returnBuffer()
CMPlayer
#player
#stream
CM_Stream
CM_PlayListStream
CM_PlayStream
#frozen
MpegFrozenFrame
#m_pFrozen
CM_MpegPlayListStream
CM_MpegPlayStream
#stream
-fastStream
MpegFast
MpegFastForward
MpegFastReverse
MpegSplicer
#m_pSplicer
#m_pPauseStream
MpegPause

FIG. 42

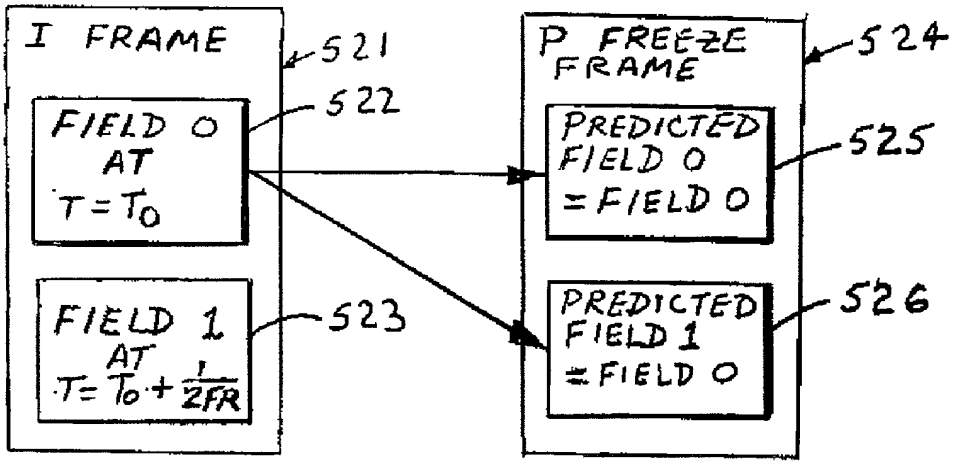
I FRAME
521
FIELD 0
AT
T = T0
522
FIELD 1
AT
T = T0 + 1/2FR
523
P FREEZE FRAME
524
PREDICTED FIELD 0
= FIELD 0
525
PREDICTED FIELD 1
= FIELD 0
526

FIG. 43

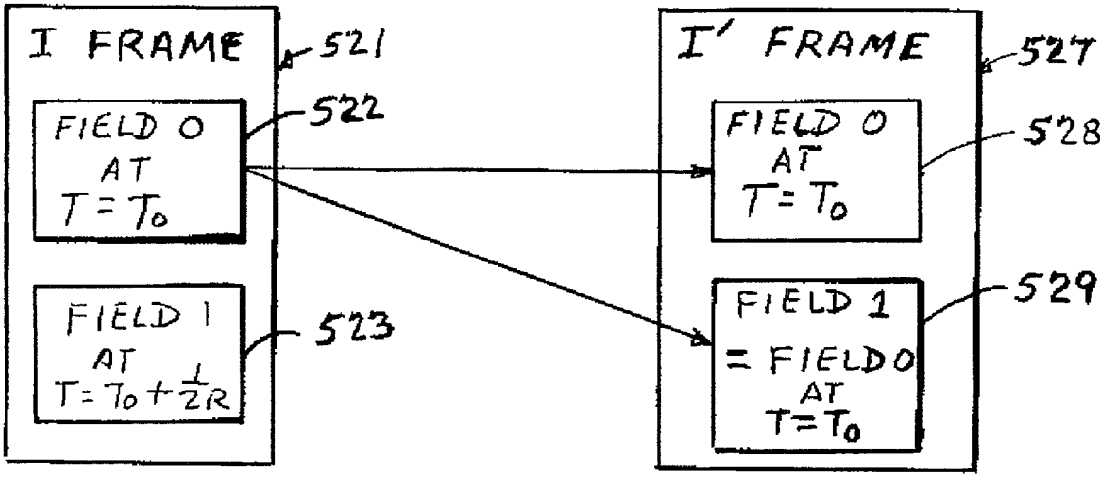
I FRAME
521
FIELD 0
AT
T = T0
522
FIELD 1
AT
T = T0 + 1/2R
523
I' FRAME
527
FIELD 0
AT
T = T0
528
FIELD 1
= FIELD 0
AT
T = T0
529

FIG. 44

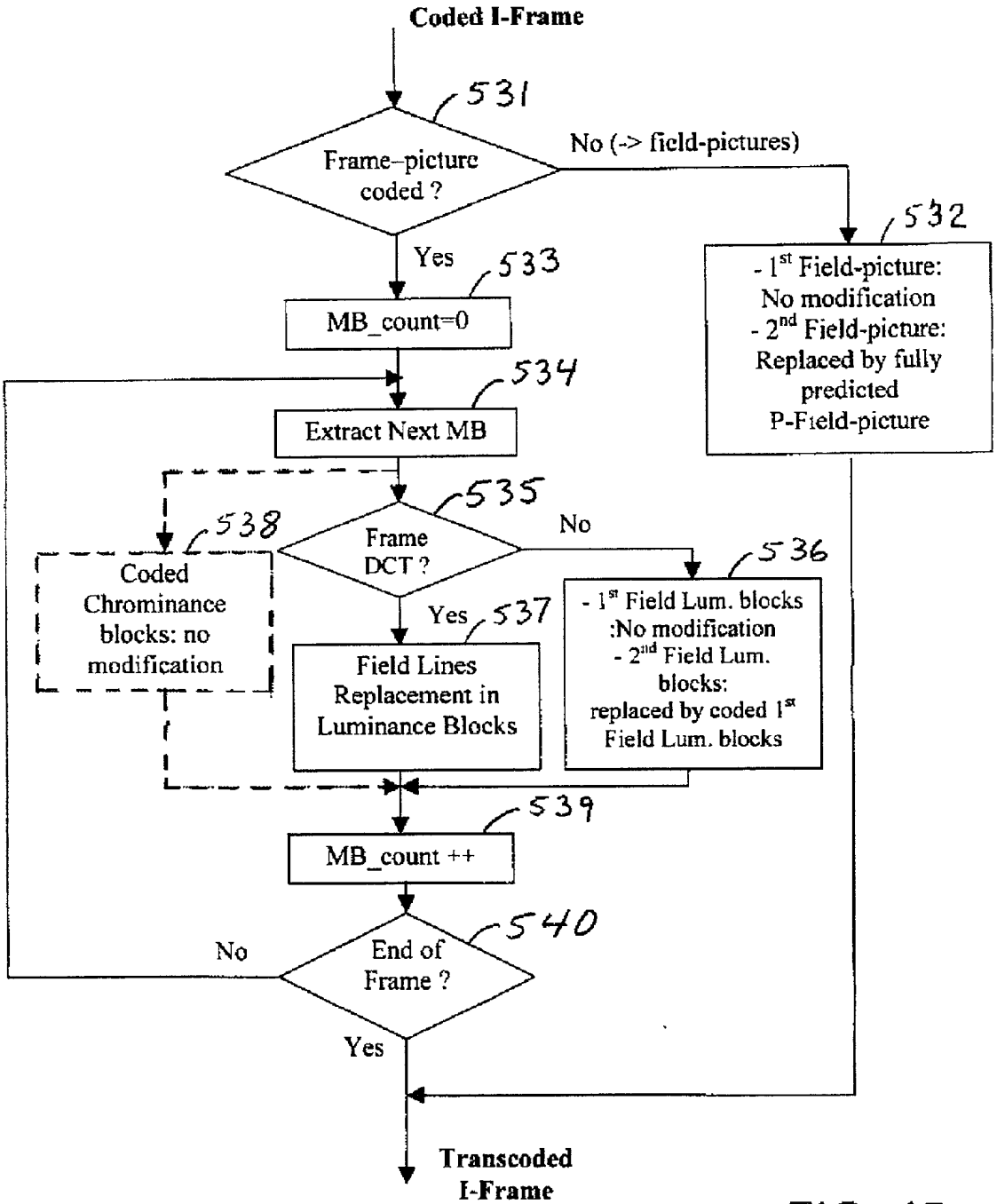
Coded I-Frame
531
Frame–picture coded ?
No (-> field-pictures)
532
- 1st Field-picture: No modification
- 2nd Field-picture: Replaced by fully predicted P-Field-picture
Yes
533
MB_count=0
534
Extract Next MB
535
Frame DCT ?
No
536
- 1st Field Lum. blocks :No modification
- 2nd Field Lum. blocks: replaced by coded 1st Field Lum. blocks
538
Coded Chrominance blocks: no modification
Yes
537
Field Lines Replacement in Luminance Blocks
539
MB_count ++
540
End of Frame ?
No
Yes
Transcoded I-Frame

FIG. 45

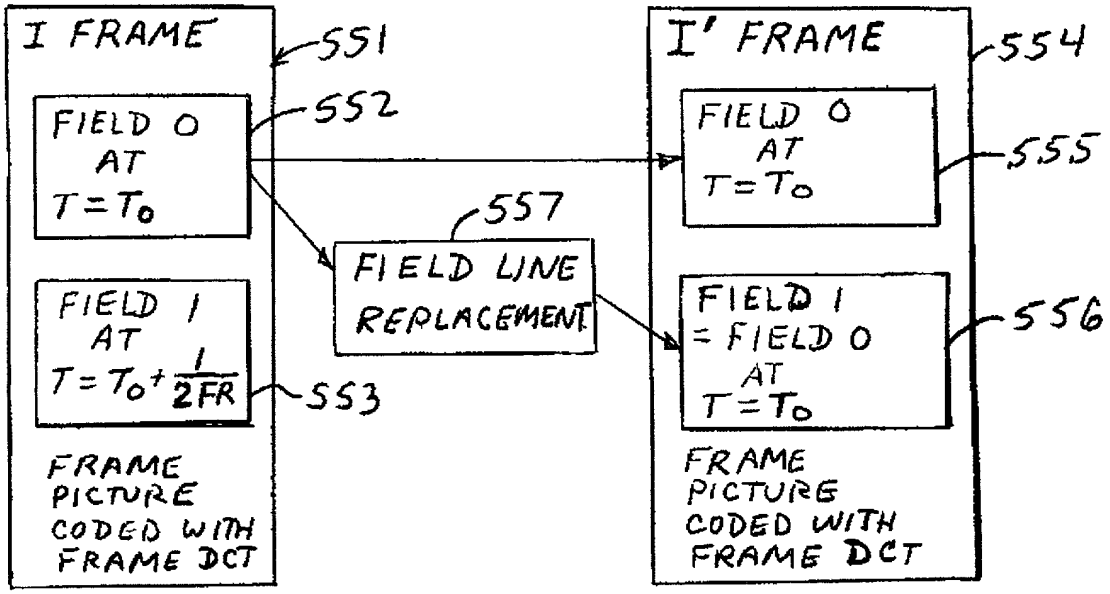

FIG. 46

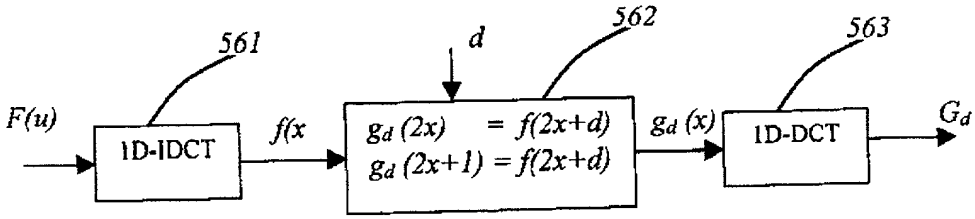

FIG. 47

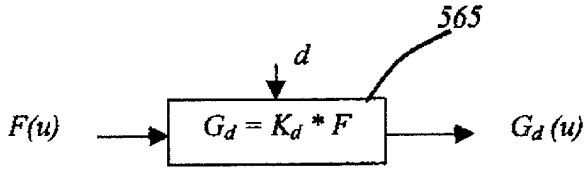

FIG. 48

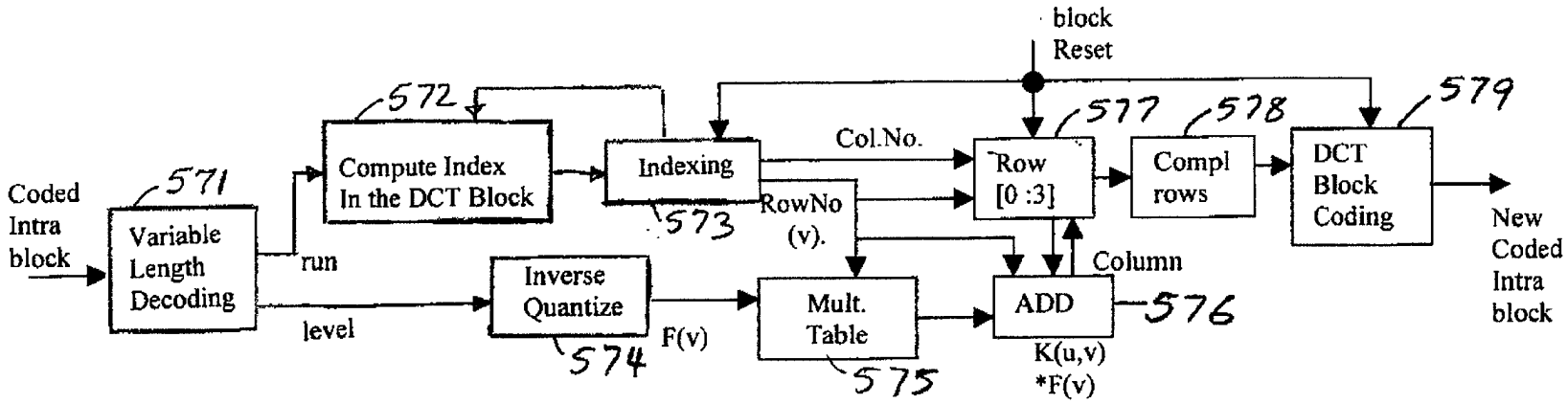

FIG. 49

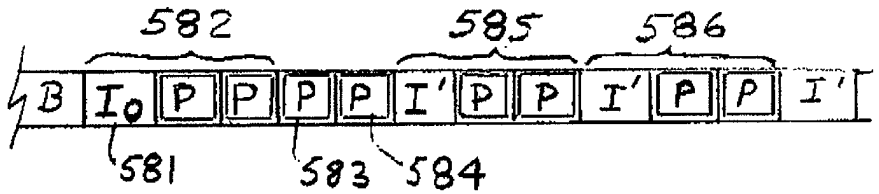

FIG. 50

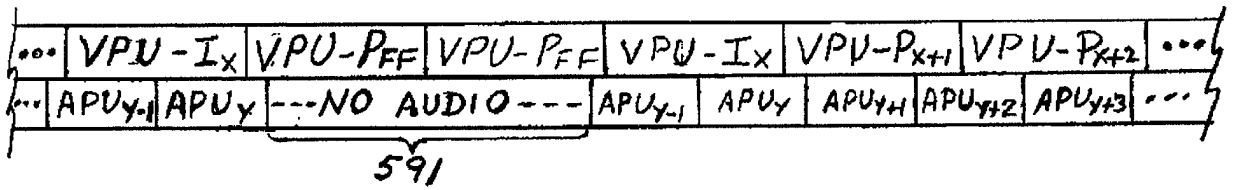

FIG. 52

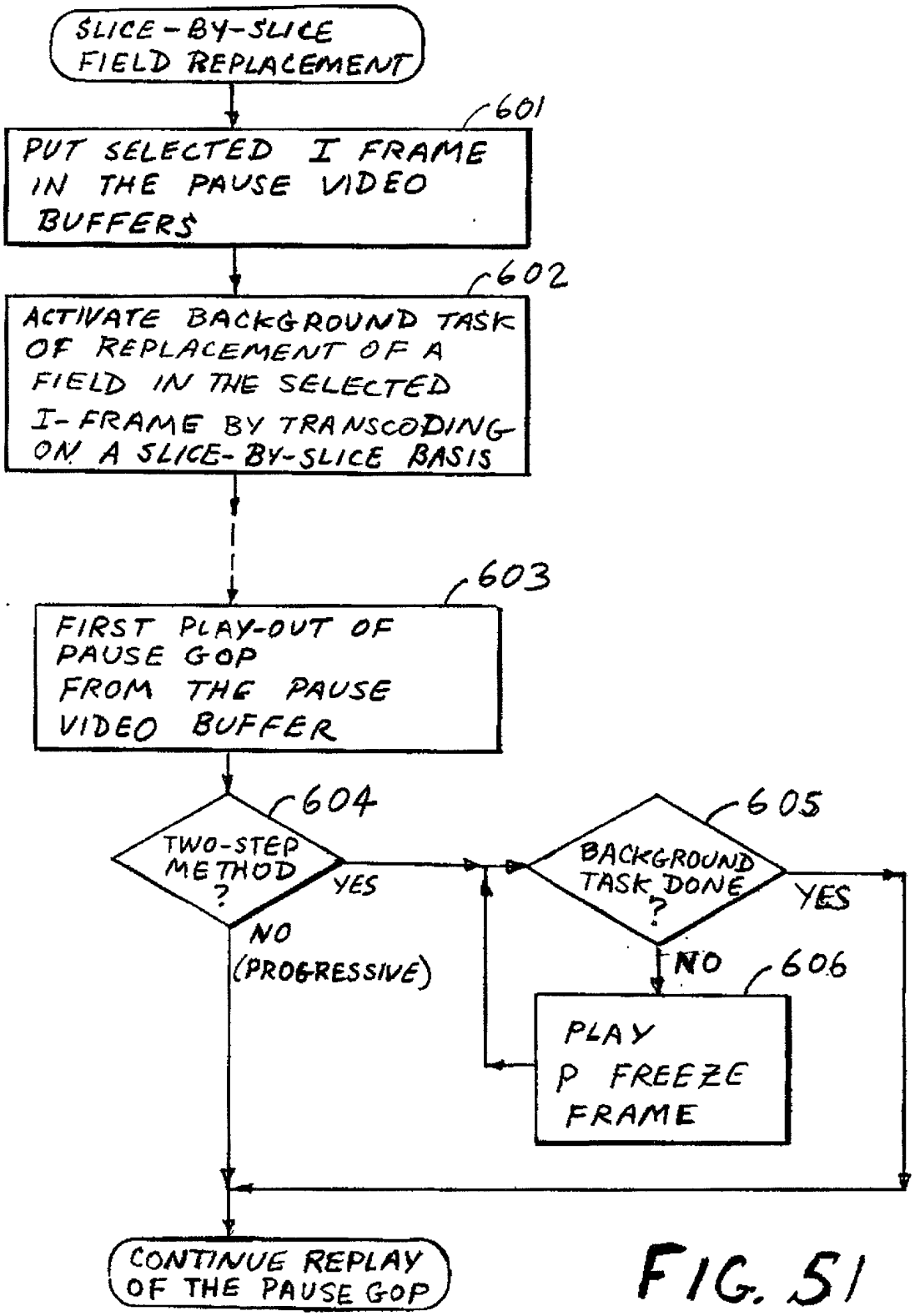
SLICE-BY-SLICE FIELD REPLACEMENT
601
PUT SELECTED I FRAME IN THE PAUSE VIDEO BUFFERS
602
ACTIVATE BACKGROUND TASK OF REPLACEMENT OF A FIELD IN THE SELECTED I-FRAME BY TRANSCODING ON A SLICE-BY-SLICE BASIS
603
FIRST PLAY-OUT OF PAUSE GOP FROM THE PAUSE VIDEO BUFFER
604
TWO-STEP METHOD ?
YES
NO (PROGRESSIVE)
605
BACKGROUND TASK DONE ?
YES
NO
606
PLAY P FREEZE FRAME
CONTINUE REPLAY OF THE PAUSE GOP

FIG. 51